Analysis of car body structures

Page, Laurence J.. 2017. “Analysis of Car Body Structures”. figshare. https://hdl.handle.net/2134/26808.

ANALYSIS OF CAR BODY STRUCTURES.

by L. J. Page. B.Tech

A Doctoral Thesis
Submitted in partial fulfilment of the requirements for
the award of Doctor of Philosophy of
Loughborough University of Technology.

October 1982

Summary.

The requirement to develop lighter vehicle structures arose as a result of the rapidly rising price of oil. The weight of a vehicle makes a considerable contribution to the power required to propel it and therefore the quantity of fuel used.

The work presented here is an investigtion into the analysis of the components of a vehicle structure, with the aim of obtaining a greater understanding of their behaviour. This knowledge is then applicable to the design of lighter structures made from an assembly of the components studied.

Analyses were undertaken both by analytical means and by the use of the finite element method. The theoretical studies were substantiated by experimental work where this was feasible. One aspect of the investigation was concerned with the comparison of fabrication techniques used in automotive structures. Here the behaviour of spotwelds, rivets and adhesives were analysed and compared. This part of the investigation was instigated by a report which suggested an increase in stiffness of 100% when adhesives were used instead of spotwelds; such a large increase was not revealed by this author's experiments. It has been shown that spotwelds can achieve 80% of the theoretical maximum stiffness.

A discussion is included on the level of confidence in an analysis for a particular finite element mesh density. Following on from this, a study of the structural effectiveness of each part of the body was made by analysing the whole structure with individual components removed.

The design and implementation of a laboratory computer to run a load-v-displacement experiment on a vehicle joint is also discussed, together with the inclusion of such joint data into the overall body analysis model.

Acknowledgements.

The author gratefully acknowledges the help and advice given by his supervisors in this project:

Mr. A. Ancliff, B.L. Technology Ltd., Gaydon.

Dr. R. Ali, Loughborough University.

Dr. P.W. Sharman, Loughborough University.

and to the Science Research Council for its financial support.

Thanks is also due to Mr.G.P. Gerrard and Mr.M.J. Hunt for their help with the mainframe computer facilities and Mr.P.L. Adcock and Mr.E. Smith for their aid with the Micro-computer and to the technicians for their help in building the experimental rigs.

Contents.

Summary..i

Acknowledgements..ii

Table of Contents..iii

Index of Figures..iv

Index of Tables and Plates................................vi

Notation...vii

Chapter 1. Introduction and Literature survey.............1

Chapter 2. Parametric study of mesh density..............12

Chapter 3. Parametric study of vehicle stiffness..........28

Chapter 4. Experimental measurement of beam stiffnesses...54

Chapter 5. Shear lag analysis of Adhesive lap joints......73

Chapter 6. Numerical solution of St.Venents equations.....90

Chapter 7. Experimental measurement of vehicle joint.....109

Chapter 8. Development of Laboratory computer............126

References...139

Appendix 1. St.Venent Torsion Program....................144

Appendix 2. Finite element mesh generator................149

Appendix 3. Micro-computer graphics routines.............156

Appendix 4. Micro-computer external control routines......161

Appendix 5. Micro-computer specification.................164

Appendix 6. Oxford polytechnic beam stiffness results 166

Appendix 7. Replacing a shear panel with bars............169

Appendix 8. Program to convert flexibility matrix to stiffness matrix...........................173

Appendix 9. Program for joint measurement experiment......174

Appendix 10. Two useful routines for PAFEC on Prime........175

Appendix 11. Program to write out PAFEC stiffness matrices.176

Index of Figures.

Figure	Title
1.1	Histogram of vehicle bending stiffnesses.
1.2	Histogram of vehicle torsional stiffnsses.
2.1	PAFEC75 level 1 elements.
2.2	ECV2A Finite element mesh skeleton.
2.3	Roof meshes.
2.4	Graph of deflections with different roof meshes.
2.5	Floor meshes.
2.6	Deflection variations with changes in floor panel.
3.1	Wavefront path
3.2	Complete model - deflection plot
3.3	Offset beam model - deflection plot
3.4	BC Post removed - deflection plot
3.5	Roof removed - deflection plot
3.6	E-Post removed - deflection plot
3.7	Sill Offset - deflection plot
3.8	Sill removed - deflection plot
3.9	Front header removed - deflection plot
3.10	Tunnel removed - deflection plot
3.11	Windscreen pillar removed - deflection plot
3.12	Front longitudinal removed - deflection plot
3.13	Lower A-Post removed - deflection plot
3.14	D-Post removed - deflection plot
3.15	Floor panels removed - deflection plot
3.16	Heel board removed - deflection plot
3.17	Rear floor removed - deflection plot
3.18	Cross beams added to floor
3.19	Histogram of stiffness and weight changes.
4.1	Oxford polytechnic beam cross-section.
4.2	Experimental beam cross-section (Loughborough).
4.3	Experimental layout for beam stiffness - bending.
4.4	Experimental layout for beam stiffness - torsion.
4.5	Shear flow path.

Index of figures continued.

4.6 Photo-stress result plots, spotweld and adhesive
4.7 Stress-strain thickness curve for birefringent coating
5.1 Lap joint geometry
5.2 Section of lap joint
5.3 Shear loading in butt joint
5.4 Butt joint geometry
5.5 Stress distribution
5.6 Plot of stress in joint
5.7 Beam geometry
5.8 Section of beam
5.9 Carpet plot of stress
6.1 Shear flow
6.2 Moment due to shear flow
6.3 Energy absorbed due to shear stress.
6.4 Shear flow path in spotwelded beam.
6.5 St.Venent Torsion Assumptions.
6.6 Absolute shear stress calculation.
6.7 Finite element mesh on beam cross-section.
6.8 Graph of torsion constant against G ratio.
6.9 PAFEC plot of stress contours on section.
6.10 plot of glue stress v distance
7.1 Joint model
7.2 T-Joint model
7.3 Joint represented by springs
7.4 Idealised joint freedoms.
7.5 Equilibrium and load matrices for joint.
7.6 End diaphragm fittings.
7.7 Spotweld analysis
8.1 Schematic representation of computer.
8.2 Drawing of simplified joint F.E. model
8.3 Program experiment - Typical output.

Index of figures continued.

Ap1.1 Cross sections for analytical St.Venent solutions.
Ap1.2 Stress plots for square bar.
Ap2.1 Digitiser 'Menu's'
Ap2.2 2D Axes definition.
Ap2.3 3D Axes definition.
Ap2.4 Terminal layout.
Ap7.1 Bar replacement for shear panels.

Index of plates.

Plate 1. Beam stiffness experimental rig.
Plate 2. Photostress analysis of beam.
Plate 3. Joint Stiffness experimental rig.
Plate 4. Micro-computer and joint stiffness rig.

Index of Tables.

2.1 Force changes around A-post with different roof meshes
3.1 Parametric study results.
4.1 Beam stiffness results.
4.2 Results of previous workers
6.1 J values for a beam, by St.Venent Torsion.

Notation.

A	Area
E	Youngs Modulus
e	Eccentricity
G	Modulus of Rigidity
g	Gravitational constant (9.81m/s)
I	Second moment of area
J	Torsion constant
K	Stiffness
k	Integration constant
L	Total Length
dp	Circumferential segment length
q	Shear flow
r	radius
T	torque
t	Thickness
tg	Glue Thickness
U	Strain energy
u,v,w	Displacements (x,y,z directions)
x,y,z	Local Coordinates
X,Y,Z	Global coordinates
γ	Shear Strain
θ , $\dot{\theta}$	Rotation angle, per unit length
τ	Shear Stress
σ	Direct Stress
Φ	Stress Function
ω	Unit Warping Function
ϵ	Direct Strain

Introduction.

The work presented in this thesis is an investigation into the analysis of components of vehicle structures with the aim of obtaining a greater understanding of their behaviour. The studies were sponsored by B.L.Technology Ltd. as part of their development of a prototype Energy Conservation Vehicle (ECV). Initially analyses were intended to be based on the whole vehicle, but it was soon found that the finite element method was unable to analyse localised joint effects as part of a complete vehicle. Further work, therefore, had to be directed at the analysis of the major beam to beam joints and also the connections between the panels making up the beams, in more detail. This knowledge is then applicable to the analysis and development of lighter complete vehicle structures.

The demand for lighter vehicles has been high since the oil crisis of 1974. Weight reduction and improved aerodynamics are the two major means of improving the fuel economy of a vehicle. Under steady state conditions the aerodynamic forces have the greatest effect, though the weight has a small effect on the rolling resistance of the tyres:

$$\text{Drag} = 1/2\ C_d\ \rho\ A\ v^2 + mg(A_d + B_d\ v)$$

ρ = air density

A = frontal area

Cd= drag coefficient

v= velocity

mg= weight

Ad= rolling res. coef.

Bd= rolling velocity coef.

Under steady state conditions weight reduction has very little effect on economy but when accelerating or climbing a hill it has an increasingly greater effect. A considerable amount of research has been done by all the major car manufacturers to decrease the weight of their vehicles. This is often done by the substitution of lighter and more efficient materials (plastics, light alloys or high strength steels). Aluminium is a common substitute for mild steel, but it has two major drawbacks:

1) The cost of aluminium is considerably higher than mild steel and

2) Its fabrication by spotwelds is less reliable and more costly since aluminium is a better conductor of both electricity and heat than steel and therefore requires a greater welding current. The presence of surface oxides also means that some surface preparation is required.

For these reasons manufacturers are taking a careful look at the use of adhesives to assemble complete vehicles. With this in mind the analysis of the stresses present in adhesives is discussed in Chapter 5.

In highly stressed areas it is possible to use HSLA (High Strength Low Alloy) steels, which are cheaper than aluminium and can be used where slender components are required such as roof pillars. Very little work has been done on the use of different materials in the same body and adhesives may prove problematical when used with mixed materials due to the differential expansion rates which could cause fractures in the brittle epoxy resins.

Car body shells are designed to a set of criteria requiring a minimum (within limits) torsional and bending stiffness for the whole vehicle, whilst staying within certain local displacement bounds when loaded (such as door aperture distortions or suspension mounting movements). Car body torsional stiffnesses vary from as low as 1600Nm/deg up to 15000Nm/deg. Figure 1.1 shows a histogram of the frequency of occurance of various stiffnesses of vehicles imported into Britain between 1965 and 1973.

To the author's knowledge no vehicle has ever been attributed poor handling characteristics due to excess flexibility. In a conversation with Dr.Macaulay at MIRA he recalled only one situation where flexibility had been a problem with respect to handling and this was a local stiffness problem on the front suspension mountings. For the vehicle involved in the work in this thesis, the co-operating company (BL Technology) set a lower limit on the torsional stiffness of 6500Nm/deg. This is a purely arbitrary value based on the experience gained from previous vehicles manufactured by the company.

Little work has been done to investigate the accuracy of a finite element model of a car with varying mesh density. What is the most coarse mesh which can be used to give results of the required reliability? This is a difficult problem to answer with any conviction as the different types of load applied to a structure can require different mesh distributions for a similar accuracy in the results. To study this problem different panels of a vehicle were modelled with a varying density of elements and a minimum acceptable density was found for each. This survey is only relevent to displacement and stiffness measurements. When stresses are required a considerably more refined mesh is likely to be required in all areas. Stresses will only, at present, be representative in the central area of panels and in larger beams.

Following from this an analysis was undertaken to find the effectiveness of each part of the structure of a vehicle on the overall stiffness. For this a second model was used based on the findings of the previous work. The method used was to completely remove each part of the structure individually and then recalculate the stiffness. The resulting information was used as an initial indicator of the parts of the structure which should be included in an optimisation study. The study itself did not act directly as an optimiser since it was too costly to make the necessary program runs to obtain any trends. To produce just the first derivatives (of the total stiffness with respect to a change in stiffness of a single component of the structure) in an optimisation study requires n+1 runs, where n is the number of components in the structure, therefore even the simplest analysis would require at least twenty runs of the program to obtain only one set of first derivatives.

Chapter 1 - Introduction.

When analysing a vehicle using the finite element method it is no suprise to find the deflections predicted are only approximately half of those found from experimental work. This is due to the effect (or lack of effectiveness) of the joints between the beams. To account for the joints in a reasonably economical way they are normally included in a finite element analysis as substructures. Joints are complex to analyse, being made up of large panels and many smaller pressings all tacked together with spotwelds. Since they have such a large effect on the stiffness of a vehicle it is very important that the analysis is representative of the real joint. To investigate this problem a joint was cut from a vehicle and its stiffness measured and compared with a finite element model. It was initially intended that the measured stiffness matrix for the joint should be substituted into a finite element model of the car, but problems with the accuracy of the measurements eventually precluded this. The necessary computer programs have been written and tested with computer models.

Due to the problems encountered in making these measurements manually, a micro-computer data acquisition system had to be developed to run the experiment a large number of times and therefore obtain statistically reliable results. The computer chosen was an S100 based micro using a Z80A processor. The use of an S100 bus system means that it is a simple matter to obtain any additional interface boards. The task for the computer was to put a joint structure through a repetitive load cycle, measuring the applied loads and deflections at six locations.

Little is known of the interaction of forces on the connections between component panels of a joint, i.e. spotwelds or adhesive. To obtain a greater understanding a series of experiments were undertaken to find the stiffness of some beams fabricated by various means (e.g. spotwelds,adhesives or rivets). This gave an indication of the differences in effectiveness of each joining method. To investigate the reason for stiffness losses a study was made of the stress distributions in different beams. This was done both analytically using finite elements and experimentally using photo-elasticity.

Lap joints were analysed using the shear lag equations for various simple loading conditions including longditudinal and transverse shear.

A modification was made to the PAFEC 75 suite of finite element programs to make the steady state temperature equations solve the St.Venant torsion equations. Many of these analytical solutions are directly comparable with the experimental measurements on the beams. The developed program is capable of combining different material properties, thus the glue line can be accurately represented and the stresses therein studied.

1.2 Literature Survey.

From the early stages of this project it became obvious that the major part of the work would have to be involved with the analysis of the joints on vehicles. Very little work has been done in this field previously, due to the complex nature of the analysis. Only since the advent of cheap computing and the introduction of the finite element method has any work of practical benefit been undertaken. Very little published work of direct relevance is available.

Considerably more literature is available for the more specific topics covered as part of the analysis of the joints. Specifically in the field of Adhesives, Finite element theory, Torsion solutions by numerical and analytical means and also Vehicle weight reduction. These topics are discussed under separate headings.

Analysis of whole vehicles by the finite element method is now commonplace and there is a considerable amount of literature available, but discussion of this large body of work would not be relevent here.

1.2.1 Analysis of Joints.

Chang (1) and Shiguta et al. (2) show in their papers the considerable effect on the overall stiffness of a vehicle that is attributable to the flexibility in the joints connecting the main skeleton beams . Chang does not, however, make any attempt to quantify the actual stiffness of a joint to gauge its effect.

Work in this field has been done by Sharman (3,4,5) in attempting to measure the stiffness of hypothetical springs located at the intersection of the two beams forming a 'T'-joint and comparing the results with simple finite element analysis. He has noted that the spring stiffness approach does not produce a satisfactory representation of the joint behaviour. Another problem with the spring approach is in relation to the position of the springs as there is no obvious point where all the beams intersect at a single point. It is unusual in a vehicle for such a point to exist since normally the beams merge together over a short length.

The finite element analyses of the beams reported by Sharman is fairly crude and takes no account of the spotwelded fabrication. This fact was found to be of vital importance in this project.

Sharman's work does show the efficacy of different stiffening measures for the chosen joint configuration, mainly by the use of diaphragms across the beam sections to stop local distortions. This study is useful in the design of stiffer joints.

1.2.2 Adhesives.

Much work has been done on this topic by Adams and his associates at Bristol University (6-10). Adhesive manufacturers do not normally quote figures for the moduli of their products, but instead they quote the UTS and peel resistance figures. Adams has tried a large number of experimental methods in an attempt to find the best method of ascertaining adhesive moduli and has found that simple test samples in tension and torsion give as reliable results as those found from adhesives specifically tested in thin film form.

Some adhesives are only available in thin film since they would overheat during curing in bulk form. For these Adams has devised two techniques:

i) The resonant bar (9). A bar of the adherend material is produced and its resonant frequency measured. The bar is then cut and re-joined using the thin film adhesive. From the change in resonant frequency it is possible to calculate the adhesive properties.

ii) Torsion test. Here a specimen of adherend is tested in an accurate torsion rig to find the adherend properties. This can now be cut in two and re-joined using the adhesive. The deflection across the adhesive layer can be measured accurately by a special rig and corrections can be made for the adherend.

The proposed adhesive for the ECV being studied in this project (Permabond ESP105) is capable of being cast into blocks (provided that care is taken to avoid overheating during curing). Samples can then simply be machined off and tested in a tensile test machine.

1.2.3 Torsion problem solutions.

The St.Venant torsion equations can only be solved analytically for very simple cross-sections (18,20,21). Timoshenko gives solutions for circular, square, triangular and elliptical bars. These analytical solutions are not of much use when considering vehicle beam cross-sections which are far too complex, necessitating the use of numerical methods.

Initial numerical solutions by other authors used the finite difference method. Zienkiewicz (19) gives solutions for rectangular bars of two materials as well as the simple check solutions with Timoshenko's results. Zienkiewicz has followed this work with a finite element solution of the same problem (13). The work in this project has been undertaken on similar lines. Another finite element solution is given by Herrmann (14) which yields good results, but is not as directly compatable with the PAFEC suite of programs as the Zienkiewicz method.

1.2.4 Vehicle weight reduction.

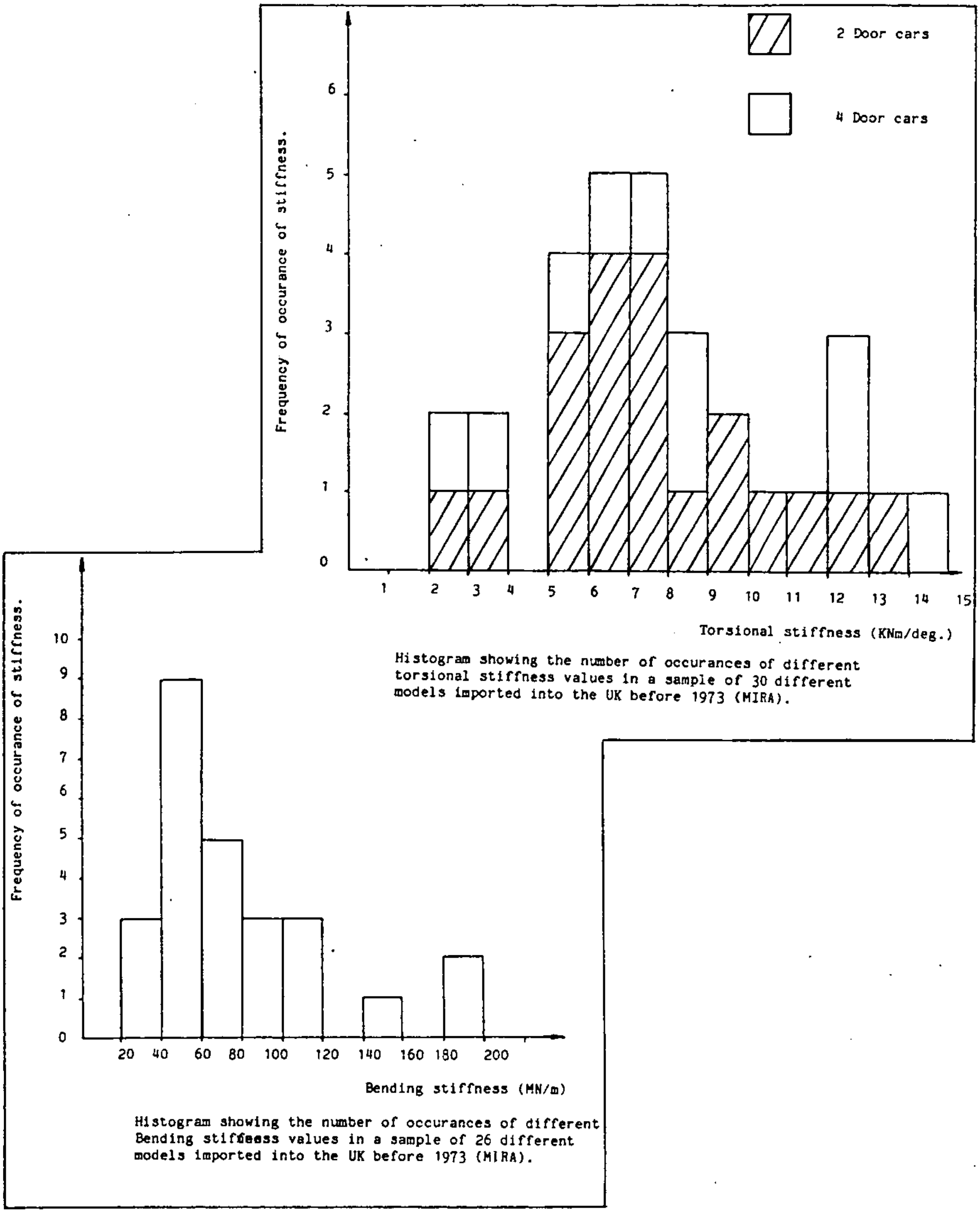
Vehicles are designed to a set of stiffness criteria, one of which is the Torsional stiffness. The actual required stiffness is difficult to define since different torsion rigs tend to give different stiffnesses for the same vehicles (22). This is assumed to be due to the different amounts of applied twist and therefore suggests that the torsion characteristics are far from linear. Thus, any finite element results are going to be, at best, approximations to the experimental tests.

Many papers have examined the use of aluminium and other more exotic materials to reduce the weight of vehicles. This varies from the use of simple substitution exercises (24,26,28,31,33,34,36) where a single panel (usually a boot lid) is under investigation, to more complex analyses where cost is taken into account (24,25,31) comparing between materials such as steel, aluminium and sheet moulding compound.

Chang in his study (25) states that the stiffness of a panel is proportional to t^{α} where α is normally between 2.0 and 3.0. In the optimisation work undertaken for this thesis a panel was analysed using finite elements with a thickness of 1 and 2mm (Figure 1.2). This showed that most of the terms in the stiffness matrix were consistent with a value of $\alpha = 1.2$. The range did stretch from 1.0 to 3.0 but very few terms had a value close to 3.0.

The structural characteristics of aluminium and design procedures for obtaining minimum weight (29,35) give an indication of the type of ribbing required to maximise the stiffness of an aluminium panel and the requirement of increased curvature to improve stiffness and denting resistance (32). The approach of different workers to the problem of weight reduction varies from empirical studies (27) through the use of finite elements on panels (29,31,32) and whole vehicles (23) to sophisticated optimisation procedures (30). These optimisation techniques are extremely expensive to run and only fairly simple structures can be analysed, such as a 30 beam vehicle skeleton.

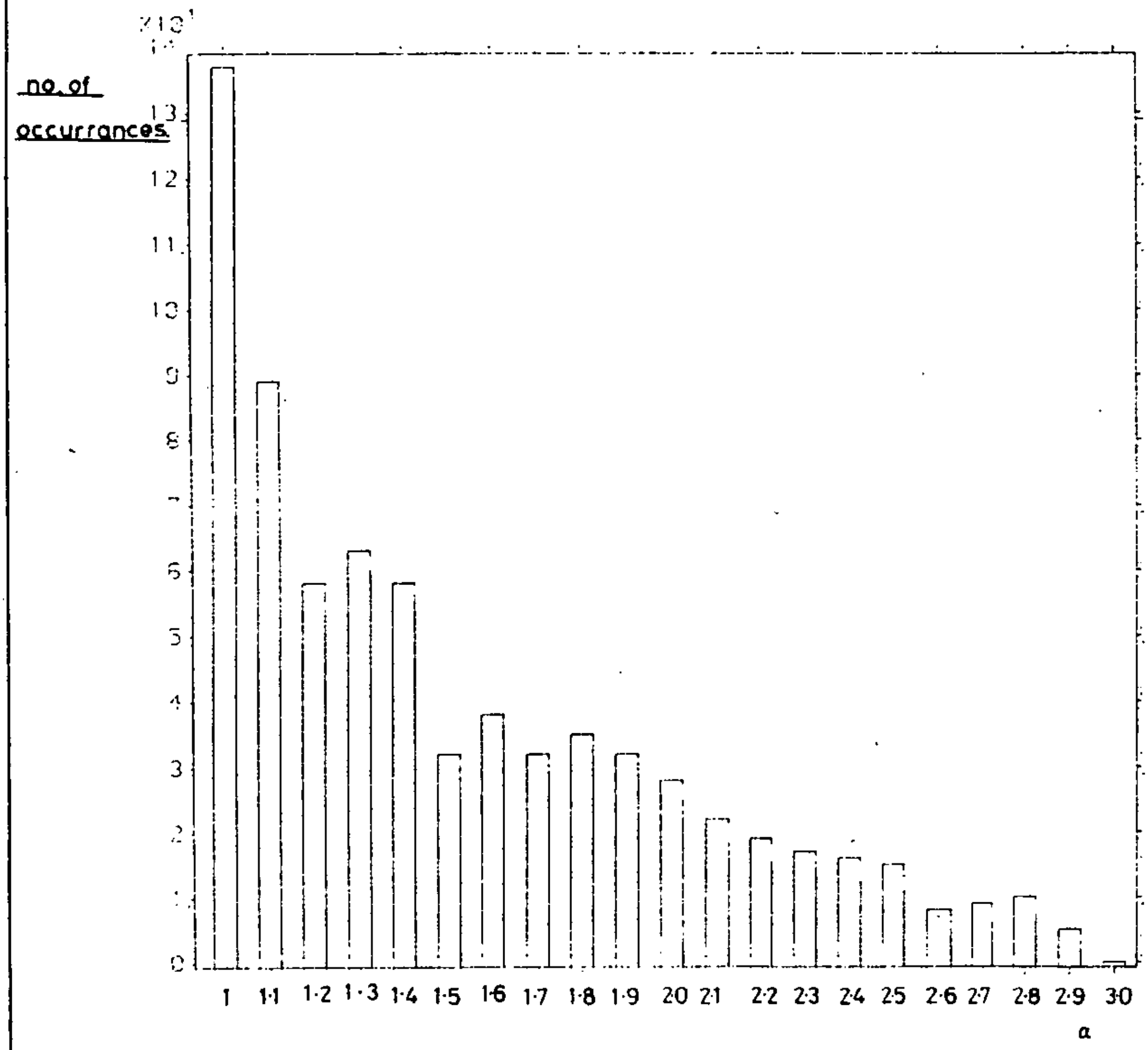
Figure 1.1 Histograms of vehicle stiffnesses



Histogram showing the number of occurances of different torsional stiffness values in a sample of 30 different models imported into the UK before 1973 (MIRA).

Histogram showing the number of occurances of different Bending stiffness values in a sample of 26 different models imported into the UK before 1973 (MIRA).

Figure 1.2 Histogram of alpha values for a panel ($K=t^{\alpha}$).



Chapter 2.

Parametric Study of Finite Element Mesh Density Requirements.

This piece of work was the first serious practical application of the Finite Element method undertaken by the author in this project. The facilities for finite element analysis were rather limited and required the use of the PAFEC75 level 1 suite of programs run with a deck of cards at Nottingham University. Level 1 was a very basic program with few error checks and uses only a subset of the modules available on later releases (Figure 2.1 shows the different element types available). A routine to produce graphical output had to be written for the computer at Loughborough to improve turn around time. Otherwise a delay of up to 3 weeks could occur before geometry errors could be found. The first model developed was extremely basic and was developed purely to obtain practical experience of the use of the program and computing facilities. The results were not considered representative and are not given here.

Having gained the necessary background knowledge of the system a more representative model was then developed (Figure 2.2). This consisted only of the beam skeleton except for the rear-quarter panel which was represented as a shear panel surrounded by stiffeners. The rear-quarter area was a particularly awkward area to model simplistically since this panel has to be included in the most basic analysis.

The first decision on the modelling accuracy must be the type of element to use. There are two types of beam element available, one with offset and one without. Is it necessary to go to the extra complexity of defining all the offset nodes, or do only some beams require offsets?

The only way to check this was to run two analyses independently using both types of element. The results obtained showed a bending stiffness reduction of 15% using offset beam elements. This is due to the bending and torsional moments induced in the joint areas by the offsets. The non-offset model is an ideal case which cannot be achieved in practice and overestimates the stiffness of the structure. The addition of offsets increases the realism of the analysis slightly for very little cost in computer time. It is therefore a worthwhile refinement in the analysis. Many of the smaller beams such as the floor stiffeners have very little effect on the overall stiffness and these need not be offset if development time for a model is limited.

How many beam elements are required in the skeleton model? No mid-span forces are to be applied externally on the skeletal frame so that only one element per span is needed. All the moments and forces that can exist are defined at the span ends. It is, however, useful to know the displaced shape at the mid spans and this requires the use of two or more elements to give the necessary nodes. This will not make any improvement to the static loading accuracy of the skeleton model at this stage, but facilitates the addition of panels later on.

During the initial stages of model development for the first Energy Conservation Vehicle (ECV), it became obvious that even the simplest modelling could not avoid the use of some panels. The particular area where the problem occurs is the rear quarter. Here two large panels make a box section over the rear wheel. This box is too deep to be represented by a beam element. The inner panel of the box is perforated and cannot therefore act as a pure shear panel. For simplicity, at this stage, two possibilities were available. Either model the area as a double panel box section or else ignore the weakened inner panel and model it as a single shear panel with peripheral stiffeners. It was found in practice that both analyses produced such a stiff structure that both could be used satisfactorily allowing the simplest to be chosen, in this case the shear panel and stiffeners.

Having decided on the basic layout for the skeletal model it was necessary to decide on the possible improvements which could be made. The main improvement was found to be the addition of panels. For each panel four different levels of mesh refinement were tried:

1) No Panel
2) Diagonal bars replace panel shear stiffness.
3) Coarse modelling using large flat triangular plates.
4) Refined mesh with curved panels.

For 2) the cross-sectional area of the bars was calculated so that they would absorb an equal amount of energy as a panel in shear. Equation 1 gives this area (a full derivation is given in appendix 7).

$$A = \frac{b.t}{4(1+v)\sin\theta\cos^2\theta} \qquad \text{Equation 1}$$

A = cross-sectional area
b and θ are given from the geometry
v = Poissons ratio
t = panel thickness

These bars are simply fitted diagonally between the corner nodes of the panel they replace. For (3) the large flat plates used were triangular elements (PAFEC 41320) fixed between the available nodes on the skeleton, with care taken to keep distortion of the elements to a minimum. The most accurate represention (4) required the addition of extra nodes to the peripheral beams to give enough connections for a relatively fine meshed panel to be fitted. It was considered particularly important to faithfully represent the curvature of the panels, especially the more sharply curved sections. For example, the roof is substantially flat but is well curved around its edges. This requires the edges to be more accurately modelled as they could add substantially to the shear stiffness of the panel and the bending stiffness of the beam.

The most refined mesh was used as a check on the accuracy of other idealisations since it bore the closest resemblance to the real panel. It would be an extremely difficult and costly process to analyse or experimentally measure the stiffness of the panels which are not simple shapes to idealise.

2.2 Parametric Study - Results.

As a starting point for this analsis the skeletal model fitted with diagonal bars was used for all panels except for the rear quarter. As the parametric study continued on each panel, the most satisfactory panel analysis for each was used thereafter.

2.2.1 Roof.

The roof was the first area of study. The remainder of the model was unchanged, using diagonal shear bars. All four levels of mesh were used as shown in Figure 2.3.

The analysis was carried out in the bending mode, with deflections taken at the central span of the sill and tunnel beams. The results are shown in Figure 2.4. As the roof model increases in complexity it can be seen that while the sill stiffness increases the tunnel stiffness decreases. This is caused by a change in the force distribution through the structure. As an example of this the forces around the top of the A-post are shown in Table 2.1 for various roof model layouts.

This shows very substantial changes in these forces. It appears that the use of diagonal bars to replace the roof is not a good representation in the bending load case. Their use causes the forces to diverge from the accurate analysis. The forces show a general decrease as the roof panel model increases in complexity.

The change in stiffness due to the removal of the roof panel is small (2% on the sill and 2.5% on the tunnel) showing that the roof panel has little effect on the vehicles overall bending stiffness. As the diagonal bars did not represent the roof well (though they did not effect the stiffness markedly) it can be assumed that the roof takes very little shear load when the vehicle is in bending. This can also be argued from the symmetry of the model under bending.

2.2.2 Floor.

A similar analysis was undertaken here, however, no attempt was made with floor panels removed or with diagonal beams. This time, the major part of the skeleton was panelled with a coarse mesh of triangular facet shell elements. This shows a marked increase in stiffness over that using bars, partly due to the use of large triangular elements which are known to be too stiff. This also suggests that the diagonal bars do not give a good representation of the panels.

Four different meshes were used to represent the floor:

(i) Coarse mesh of triangular facet shell elements.

(ii) Accurate mesh of quadrilateral elemets (PAFEC 44200).

(iii) Mesh of beams which criss-cross the floor (outriggers, etc.)

(iv) A combination of (ii) and (iii).

Figure 2.5 shows the different mesh patterns and Figure 2.6 gives a diagramatic representation of the results.

It can be seen from Figure 2.6 that the beams and panels share the job of stiffening the floor since very similar deflections occur for both models. The stiffness is approximately doubled when both are combined.

This shows it to be essential, when bending modes are present, to develop a fairly precise model of the floor, taking account of all the small beams and stiffeners. Stiffening features such as the foot wells should also be accounted for but this need not be done in too greater detail. Simply moving the nodes in the centre of the panels down to their correct height will distort the panel adequately to produce any stiffening effect.

2.2.3 Rear quarter.

This area cannot be practicably modelled using beams or bars. There are two main possibilities for a coarse mesh of shell elements to represent this area:

(i) A single panel, 1mm thick, surrounded by small stiffening beams to represent the flanges and connections to other panels.

(ii) A box section to include a representation of the (perforated) inner skin.

Both of these arrangements produced a very stiff model of this area, but neither had any different effect on the overall vehicle stiffness. Both of these representations can therefore give acceptable results. Since the single panel model is simpler (and cheaper) this was the one chosen for use here. The inner perforated beam when combined with the curved outer panel does not increase the stiffness significantly.

2.2.4 Rear wheel arch and floor.

The modelling of this area is not fully representative of the real case as the loads were being supported on the rear end of the sill, not at the suspension mountings. This was done because no design was available for the rear suspension at the time of the analysis. For this reason little can be gained from the analysis of this area. All the results do show, is that the diagonal bars are unsuccessful at representing this area and at least a coarse shell mesh is required.

2.3 Stiffness results.

The bending stiffness calculated from this finite element model in its final form is about four times that measured on the Metro. This is about eight times that which would be expected for the full ECV2A body when taking account of the fact that it is made of aluminium, albeit of a thicker gauge than the steel gauge of the Metro.

How can this increased stiffness be accounted for?

(i) A factor of about two can be taken out for the use of rigid joints between the beams. (However until further work is undertaken no better representation can be made).

(ii) The body was not supported at the suspension mountings but, instead, at the ends of the sill, reducing the effective span by about 30%. Since the stiffness of a beam is inversely proportional to its length cubed, this will account for a further factor of 3.

By these two simple calculations the stiffness can be shown to be fairly close to the expected value. Accurate accounting for (i) would involve considerable effort. Item (ii) could be accounted for better if the analyst had closer liason with the vehicle designer. But at the early stages of a design, as here, only certain parts were finalised, the rest of the design was subject to alteration from day to day.

2.4 Conclusions.

1) Much care must be taken to develop a refined modelling of the structure around the main load carrying areas, these include:

(a) Front and rear suspension mountings.

(b) The main floor (for passenger loadings) and the rear floor (if luggage is to be considered).

(c) The roof area, if roof rack loads are to be considered. Most modern roof racks are fixed to the gutter and therefore to the cantrail, and so transfer their load directly into a beam. If it is considered to be necessary, the older style rack with pads on the roof would require a fine mesh model of the roof panel.

2) Large flat panels, well away from any loaded areas need not be meshed very finely. They tend to be fairly evenly stressed and do not support large loads. These remarks apply to:

(a) The roof (under normal loading cases).

(b) The front bulkhead.

(c) The rear quarter panel(s).

3) The mesh density of a very stiff area (away from loading points) tends not to be very critical to the final overall stiffness, since a very stiff area contributes little to the flexibility of the structure.

The very stiff areas are of course those areas where greatest weight reduction can be achieved. For example the rear quarter could be reduced in weight, but only at the expense of greater susceptibility to denting.

Table 2.1 Force changes around A-post with different roof meshes.

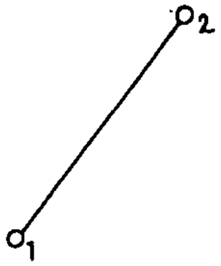
Forces in the cantrail at the connection to the A-post and Front header rail under a bending load.

Roof mesh	Axial Force	Shear (Y)	Shear (Z)	Torsion	Bending Moment Y-axis	Bending Moment Z-axis
No Roof	409	34.1	62.8	.108	-21.6	8.8
Cross Bars	417	33.3	63.7	.112	-21.9	8.6
Rough Shell	123	41.8	57.6	-.454	-11.9	12.6
Acurate Shell	242	54.1	42.6	.194	-10.1	15.4

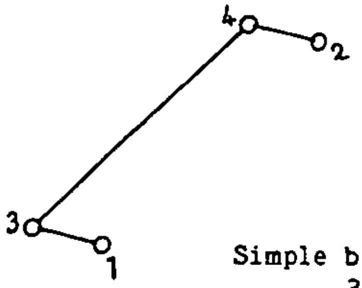
All Units in Newtons and Meters.

Forces act along the local beam axes.

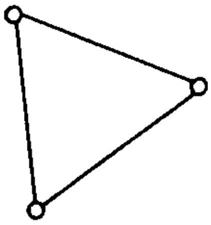
Figure 2.1 PAFEC75 level 1 elements.



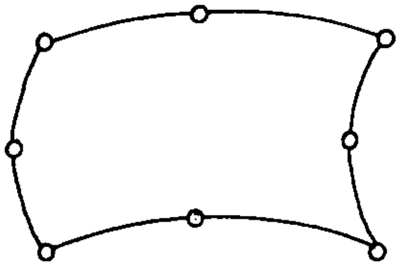
Simple beam element
34000



Simple beam with offset
34200



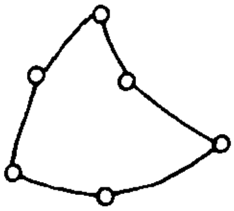
3 noded triangular shell carrying in plane and bending forces. 41320



Eight noded facet shell element carrying membrane and bending forces. 44210

Similar elements:

44110



44200

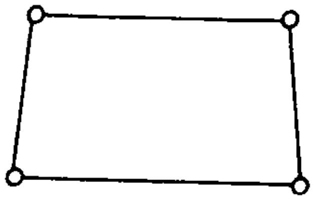


Figure 2.2 ECV2A finite element mesh skeleton.

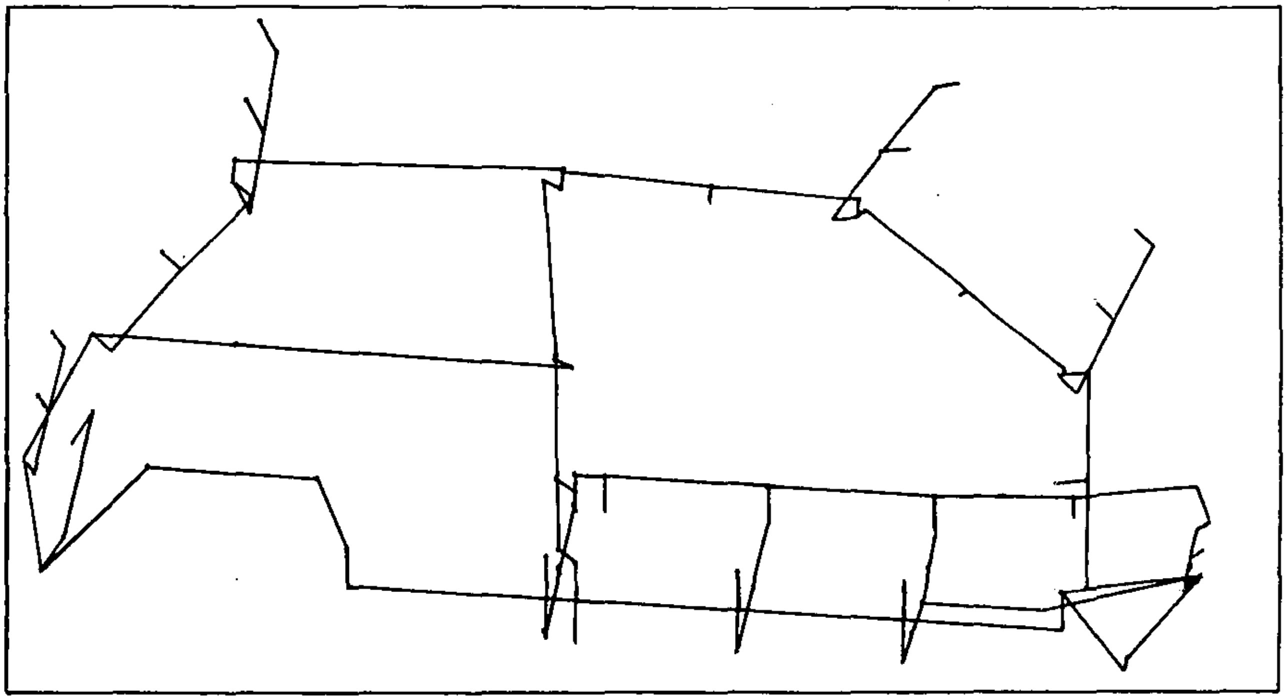
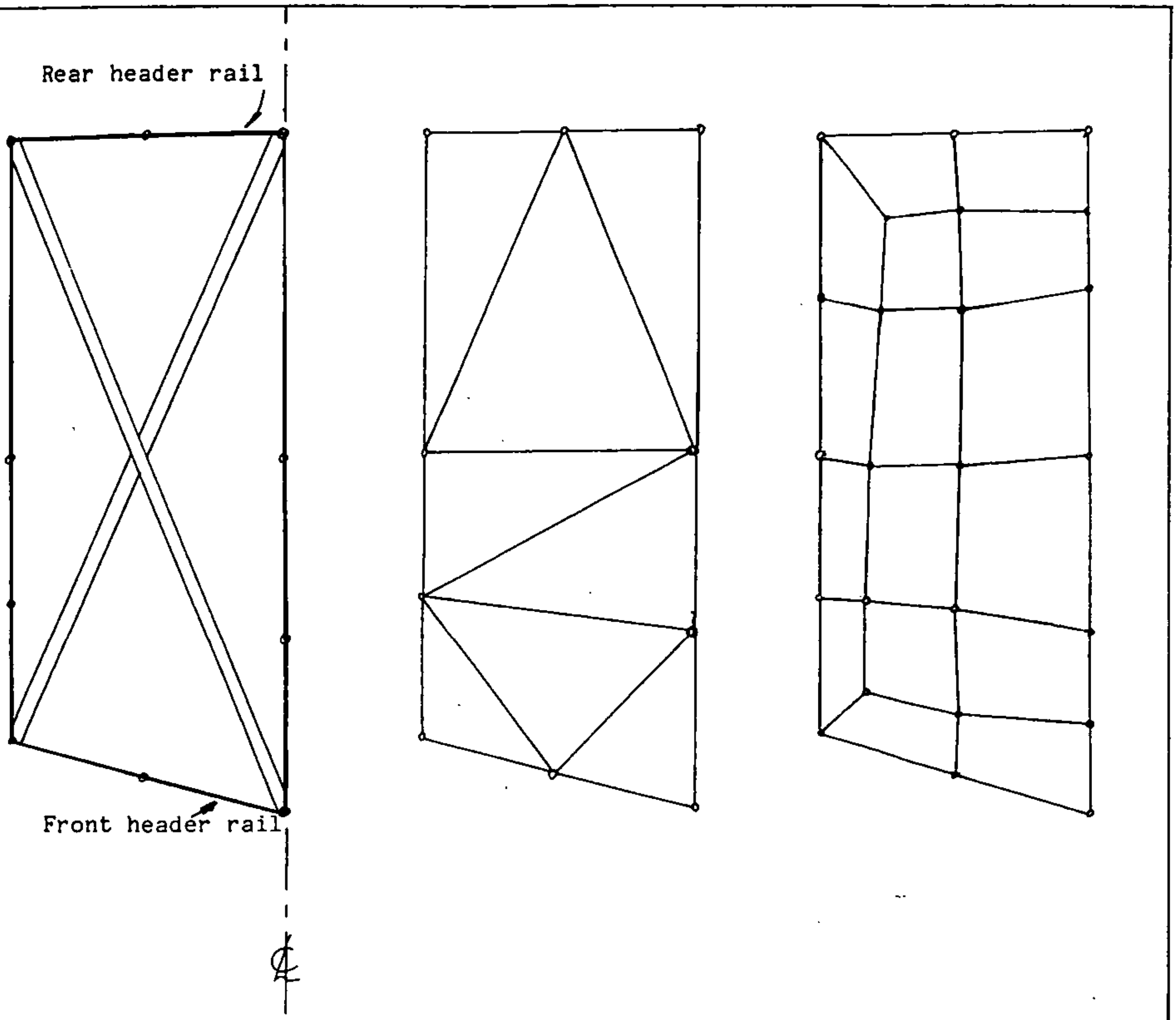


Figure 2.3 Different roof mesh refinement levels.



Diagonal bars replacing panels

Rough shell mesh.

Accurate shell mesh

Figure 2.4 Deflections with different roof meshes.

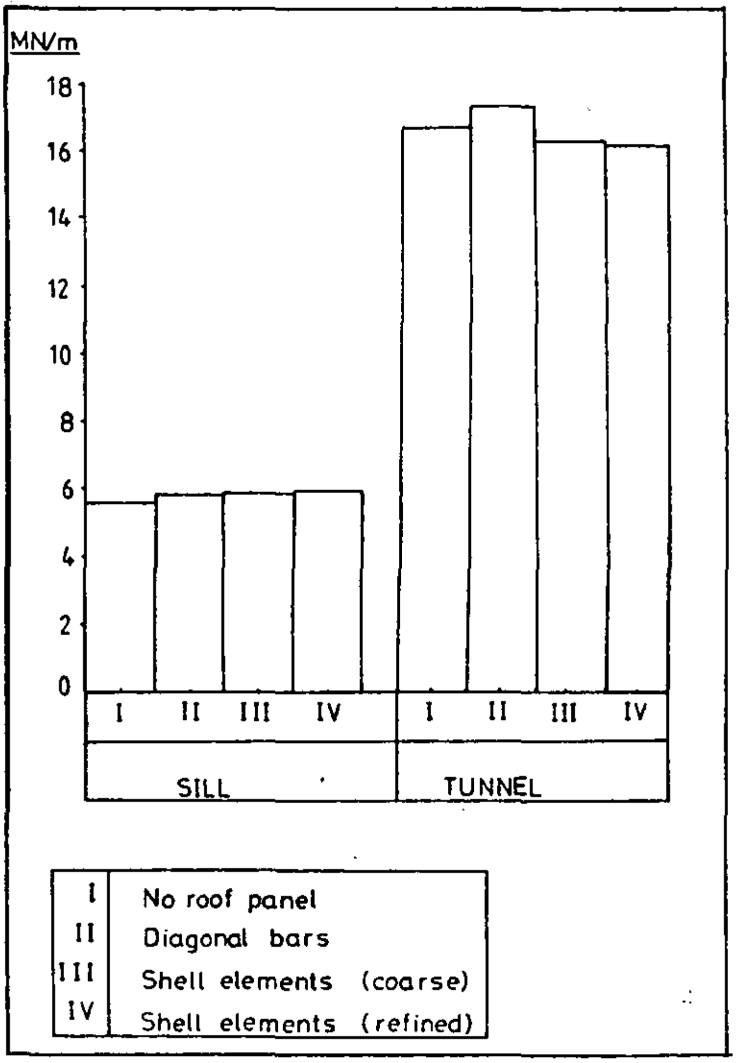


Figure 2.5 Different floor mesh refinement levels.

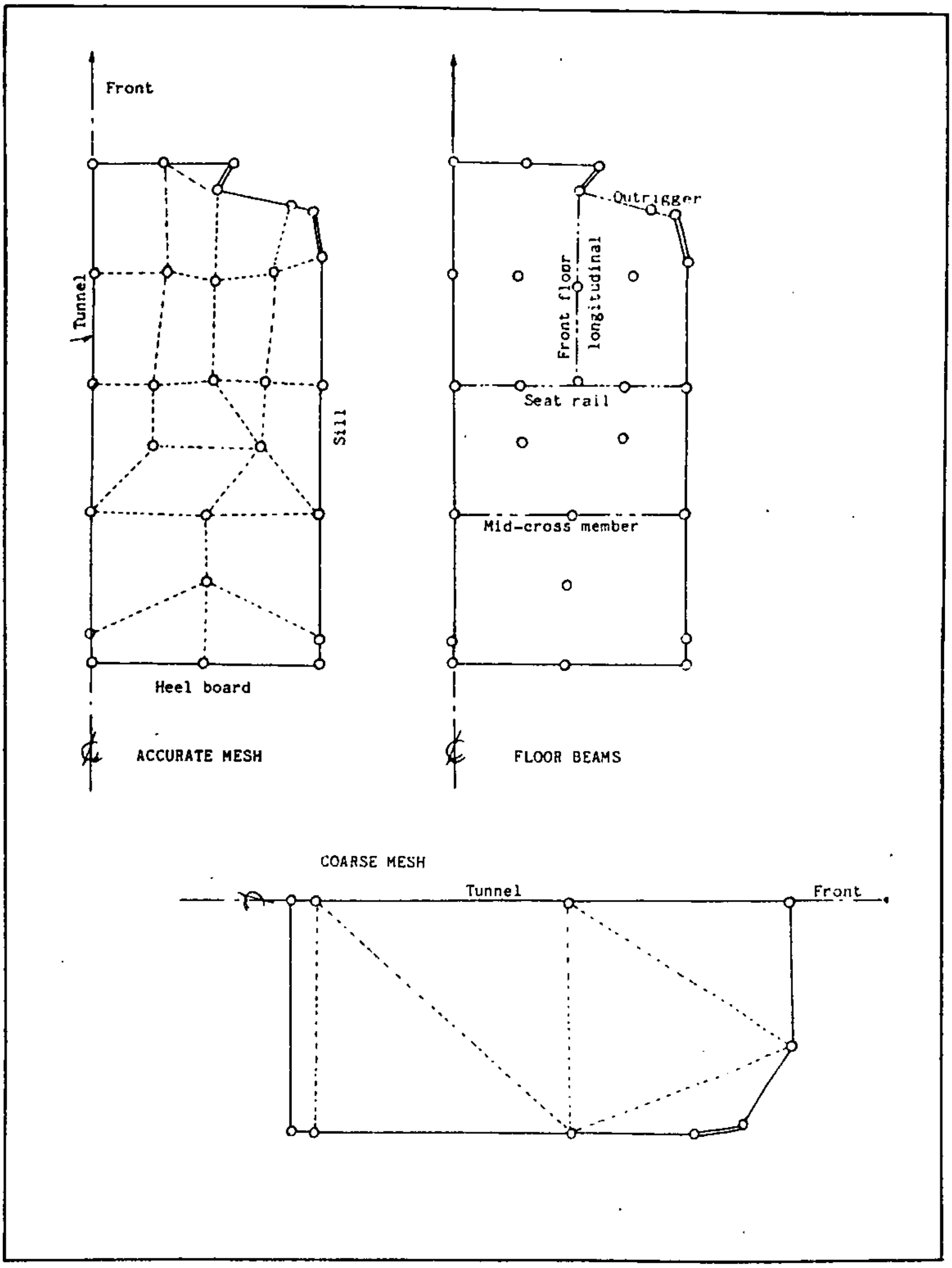
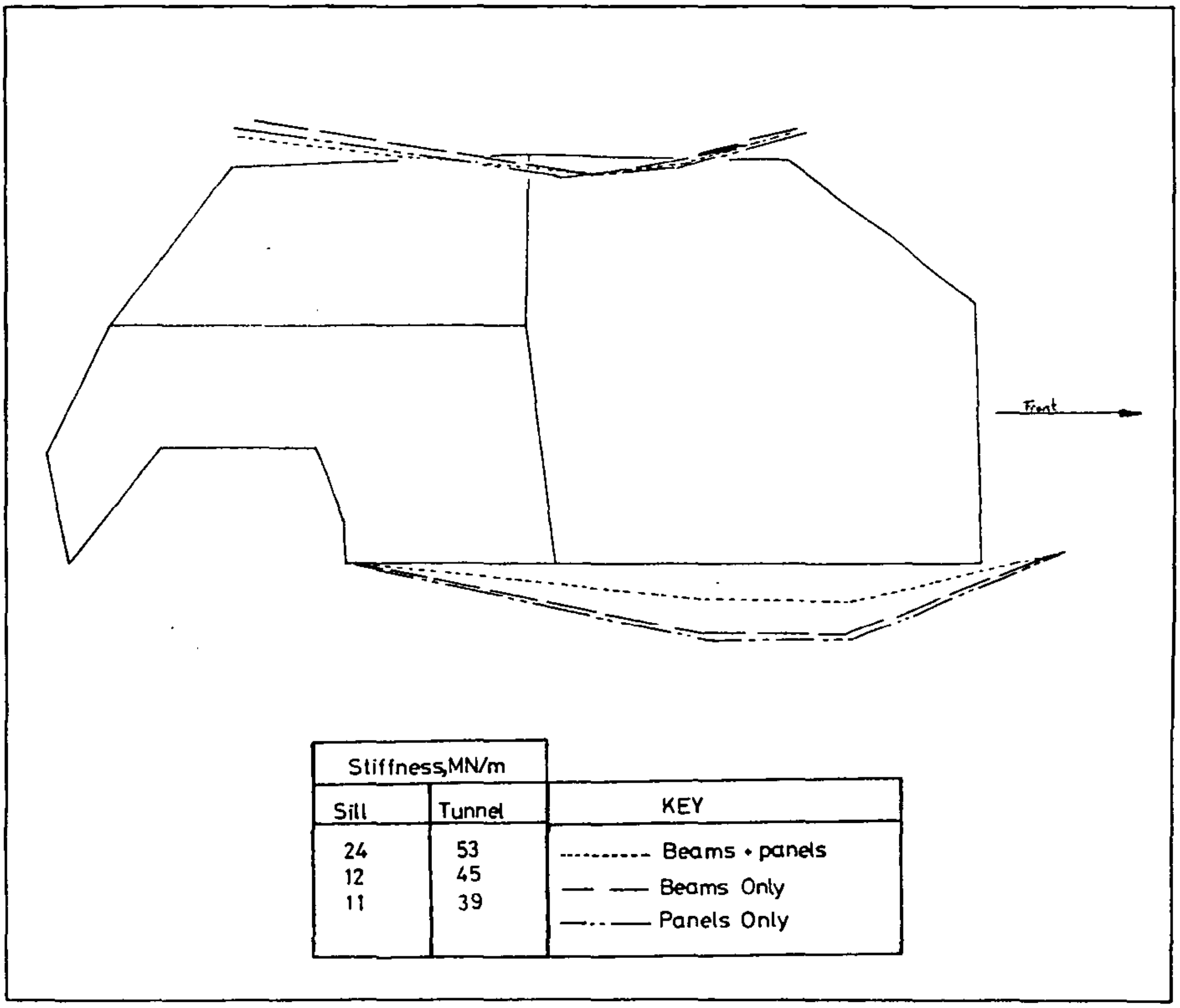


Figure 2.6 Deflection variations with different floor pan models.



Stiffness,MN/m		
Sill	Tunnel	KEY
24	53	.... Beams + panels
12	45	— — Beams Only
11	39	—·· — Panels Only

Parametric analysis of a vehicle.

The objective of this part of the work was to find the contribution made by each part of a vehicle structure to the overall stiffness of the vehicle. The method chosen was to develop a simple but comprehensive finite element model of the vehicle. This could then be analysed many times with various parts removed, to give an indication of the effectiveness of each part.

It was also hoped to develop from this a simple optimisation procedure. Some work was done on developing a computer program to do this, but it became obvious that the time involved would be too great and so this had to be abandoned at a fairly early stage.

3.1 Finite element model development.

The finite element model was to serve a number of purposes, not least as the 'first-try' skeletal model of ECV3, for B.L. Technology Ltd.. To be comprehensive the model had to meet the following requirements.

(i) The cost of a run of the model had to be kept to a minimum.

(ii) The model had to be suitable for dynamic analysis so that quite a large number of nodes had to be used to avoid an overly coarse mass discretisation.

(iii) The model had to be suitable for use with both the SDRC SuperB and the PAFEC75 suite of finite element programs.

(iv) The model should easily be converted from two door to foor door form.

(v) Provision had to be made to allow for the inclusion of joint substructures in the analysis without the need for major alterations.

3.1.1 Meeting the model requirements.

(i) This requires the wave front to be kept as small as possible. To ensure this, it is necessary to keep the number of nodes across the car low so that the wavefront can traverse back and forth keeping only a few nodes active. With 6 nodes across the width of the car only a maximum of 10 nodes will be active at any one time (a wavefront of 50-60 degrees of freedom) as shown in Figure 3.1.

(ii) The need for a reasonably large number of nodes works against the low cost requirement, but using the mesh formation as described in (i) a large number of elements can be used along the length of the car without incurring excessive cost. It must be noted that the cost of an analysis goes up with the cube of the wavefront size but only in direct proportion to the number of elements (keeping the wavefront constant).

(iii) A conflict arises between the two computer programs inasmuch as PAFEC75 can only use flat shell elements while SDRC SuperB can use curved elements. It was required that the SuperB model should be comprehensive so that the panels were curved (but only as little as possible). A simple program was written to re-order the element topologies for PAFEC75.

(iv) Though most of the model was identical for both the two and four door models it was found impractical to make the model interchangable. Instead two models were developed in parallel.

(v) To allow for the addition of joint sub-structures, an easily removable area around each joint was provided. This was done by placing a node at 1/5th of a span from the major joints. This allows the 3 or 4 beams of the joint to be simply 'commented' out and the sub-structure merged in its place.

The design of ECV3 was not complete while this model was being developed so that much of the model had to be filled in with parts of the ECV2A skeleton. The beam properties were calculated with special care since ECV3 was to be assembled with adhesive and not spotwelds. For the calculation of Iyy, Izz and Area the full cross section was used, but the torsion constant was calculated with the flanges removed (see section 5.2).

3.2 Parametric study.

Figure 3.2 shows the deflection plot of the complete model for comparison with the plots with members removed. The offset beam plot of Figure 3.3 shows that there is little change in the distortion mode so that it is unnecessary to use the extra complexity in the model.

A histogram representation of the data is given in Figure 3.19 for clarity.

3.2.1 BC post. (Figure 3.4)

The loss in stiffness when this member is removed is due mainly to the the fact that the connection to the roof is lost. This means that the roof then takes less share of the load. The tunnel deflection remains substantially unchanged.

With the roof connection removed, the roof now moves forward transferring some extra load to the windscreen pillar where it is not desireable. The deflection here is increased by 14%.

3.2.2 Roof Panels. (Figure 3.5)

The analysis given in Chapter 2 of ECV2A suggested that there would be little change in stiffness with the roof panels removed. This was in fact the case as ECV3 shows very little change in stiffness - 0.6% on the tunnel and 1.5% on the sill - in the bending mode. In torsion the change is more significant at 7.5%, which shows the effect that the roof panel has in shear.

The weight is reduced by 4% by the removal of the roof panel which leaves a net advantage in weight/stiffness ratio in the bending mode. There should therefore be some scope for lightening this panel, although for such a large panel dent resistace may become a problem. Roll over protection should be provided by the roof beams.

3.2.3 E-post.

ECV2A had exhibited a weakness in the location of the roof which tended to move fore and aft considerably. The addition of the E-post on ECV3 was intended to overcome this. The results show little change in stiffness with the E-post removed, and Figure 3.6 shows no tendency for the roof to move. It therefore appears that this beam is superfluous on ECV3 since the D-post is far better located.

The E-post accounts for 2.5% of the weight of the vehicle but only 0.5-1.0% of bending and 2.1% of torsional stiffness. It would therefore be advantageous to remove the beam altogether.

3.2.4 Sill. (Figures 3.7 and 3.8)

The sill is the main supporting member in bending and when removed the structure distorts massively in this mode. However in torsion the stiffness reduction is only 11%.

It would be unreasonable to try to reduce the weight of the sill, indeed it could be advantageous to increase its size, especially if the joints connecting it to the rest of the structure could be improved to make it carry a greater proportion of the load. Not only would this improve the vehicle static stiffness, but it would also make the vehicle safer under impact due to the stiffening of the passanger compartment.

Offsetting the sill (i.e. making the car slightly wider) will make an improvement in torsional stiffness, changing the sill position by only 50mm increased the torsional stiffness by 2.5%.

3.2.5 Tunnel. (Figure 3.10)

The tunnel supports fairly large loads in bending, thus as for the sill its removal caused excessive deflections. However in torsion the tunnel made almost no difference to stiffness. It should therefore be possible to remove the tunnel if a beam were put across the car to support the floor. A tall slim beam should be adequate. This arrangement should improve torsional stiffness but would require the sill to be enlarged for bending.

3.2.6 Front header rail. (Figure 3.9)

All the Figures show this beam to bend considerably when the vehicle is under torsion. Indeed its removal (Figure 3.9) caused a stiffness reduction of 1.5%, which is more than would be expected for a relatively insignificant beam in a non-load carrying area. The results show this beam to be subject to one of the largest rotations on the vehicle. It would probably be advantageous to make the beam a closed section, especially as distortions around the windscreen must be minimised.

3.2.7 Windscreen pillar. (Figure 3.11)

This beam should aid the transmission of some load to the roof, but its removal only reduces the torsional stiffness by 13%. More load would be transmitted if the glass were represented since its shear stiffness would prevent lateral movement of the pillar.

In bending, the displacements are similar to those produced by the removal of the BC-post. The BC-post cannot transfer much load to the roof since the cantrail simply rotates about the D-post joint. Load is only transferred by moments in the joints thus if the joints were properly represented even less load would be transferred to the roof.

3.2.8 Front Longditudinal. (Figure 3.12)

Removal of this makes little difference in bending but accounts for a 20% reduction in torsional stiffness. In practice this beam stops local distortions around the front suspension and acts as an energy absorber in head on collisions. The effects of this beam are therefore more local and no recommendation can be made from this study about weight reduction.

3.2.9 Lower A-post. (Figure 3.13)

This beam also shows a fairly favourable weight/stiffness ratio but has more effect locally than on overall stiffness. It must support the front door and must not twist excessively when the door is bent back or leant upon. Both these criteria are fairly subjective and affect the feel of quality of the car more than anything else.

3.2.10 D-post. (Figure 3.14)

This beam proves to be very important showing a very significant effect on both bending and torsional stiffness. The bending deflection mode changes in the following ways due to its removal:

(i) The front of the car falls making the windscreen pillar pull the roof down and forwards.

(ii) The rear quarter rises pushing the rear floor down with the heel board and the tunnel.

In torsion the rear of the roof is free to move laterally and therefore imparts little shear strength.

3.2.11 Floor panels. (Figure 3.15)

The floor panels contribute 12% to the weight of the vehicle and 20% to the torsional stiffness. With no support from the floor the tunnel shows exessive deflection, but if the tunnel were replaced by a cross-beam this would be no problem (Figure 18). The sill shows a decrease in deflection since it is carrying less load.

If the load can be taken elsewhere it may be possible to use alternative materials for the floor, such as a foam filled laminate which would be fairly rigid and have the added advantage of sound deadening.

3.2.12 Heel board. (Figure 3.16)

The heel board acts as a simple panel in shear to support the rear of the main floor and the tunnel. Therefore, when removed, it allows the floor to sag. Though this panel could probably be reduced in thickness it would not make a substantial contribution to weight reduction.

3.2.13 Rear floor. (figure 3.17)

The finite element model gives a large vertical deflection in the middle of the rear floor section which would not be expected. This probably occurs because (a) the area is very flexible in bending, being modelled as a flat panel and (b) local forces around the rear suspension mountings force the wheel arches, rear cross-member and heel board together and with a moment on the front lip from the bending in the floor the panel can only deflect upwards.

Removal of the panel causes large local distortions around the suspension mountings.

Conclusions.

3.3.1 Many of the beams on the car appear to be only effective in one load case, for instance the tunnel in bending only and the front longitudinal in torsion only. It should therefore be possible to modify either the beam sections or their positions so that they become effective in both modes.

3.3.2 When some beams are removed the transfer of load around the vehicle causes deflections to increase in some places but decrease in others. It may therefore be possible to reduce the properties of a beam to cause its load to be transfered elsewhere and actually reduce the deflection at the desired point, without inducing an unwanted deflection anywhere else.

3.3.3 Some panel sections carry very little load and may be replaced by lightweight plastic panels. These areas include the roof and floor as well as outer panel work such as the bonnet lid, boot lid and doors.

Figure 3.1 Wavefront path across a panel.

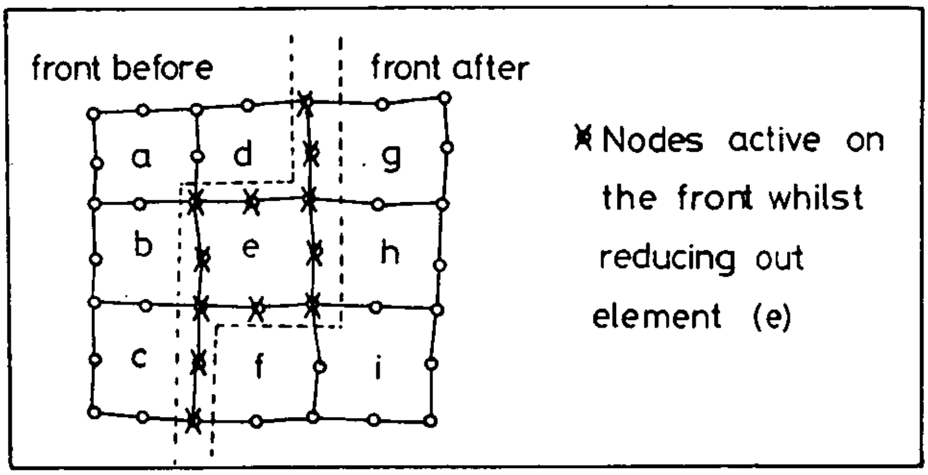


Figure 3.2 Complete model deflection plot.

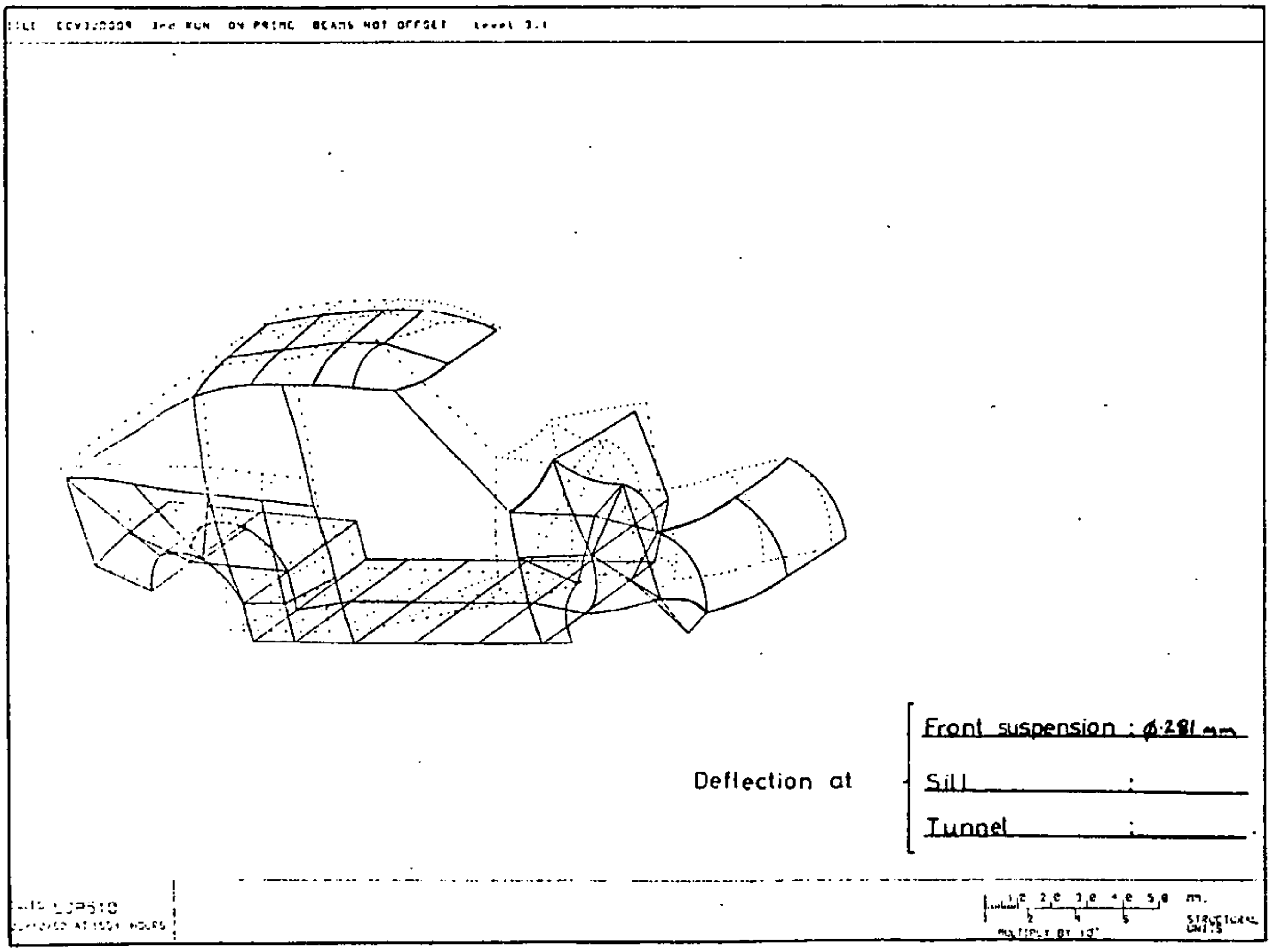


Figure 3.3 Offset beams deflection plot.

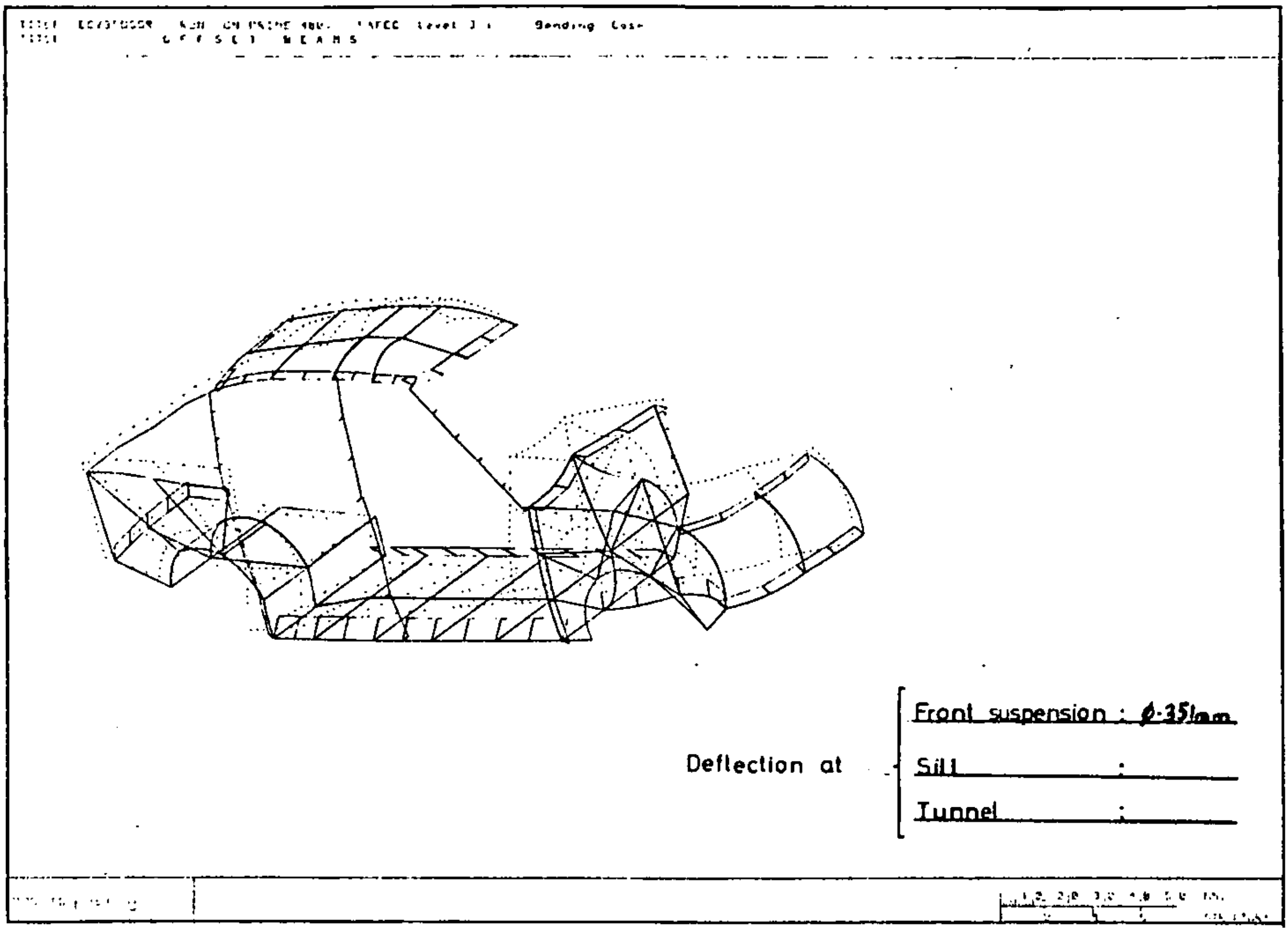


Figure 3.4 BC post removed - deflection plot.

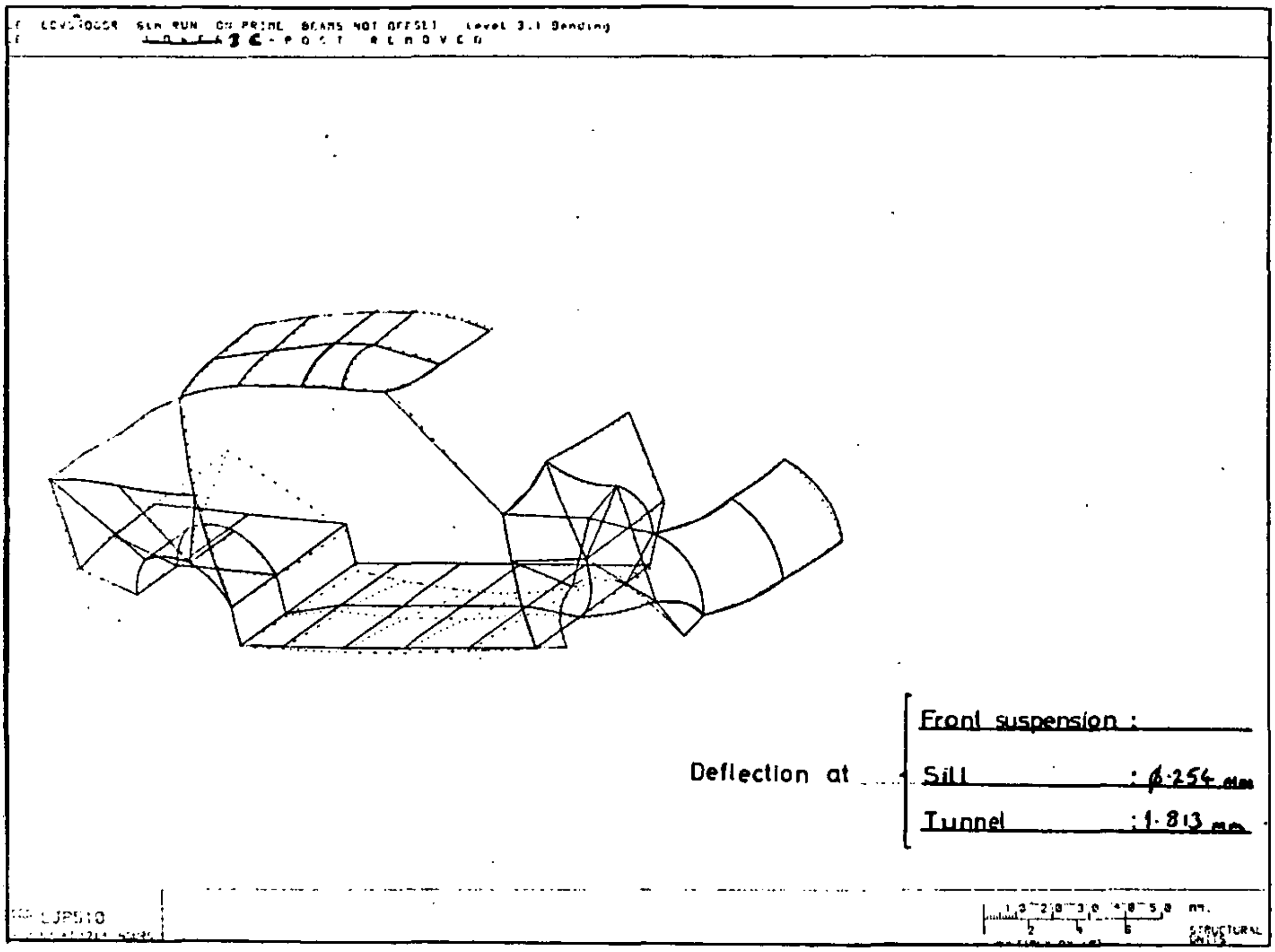
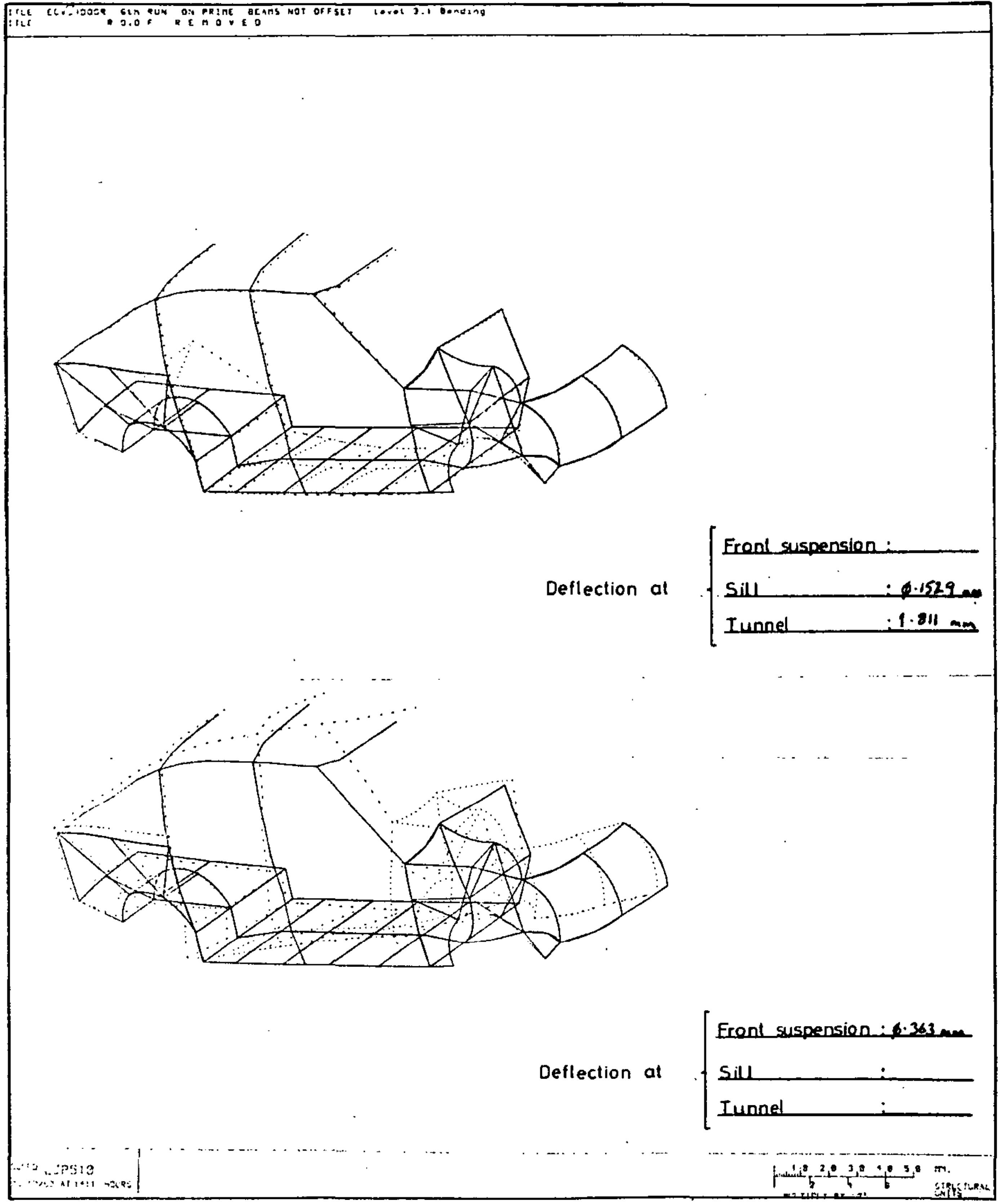


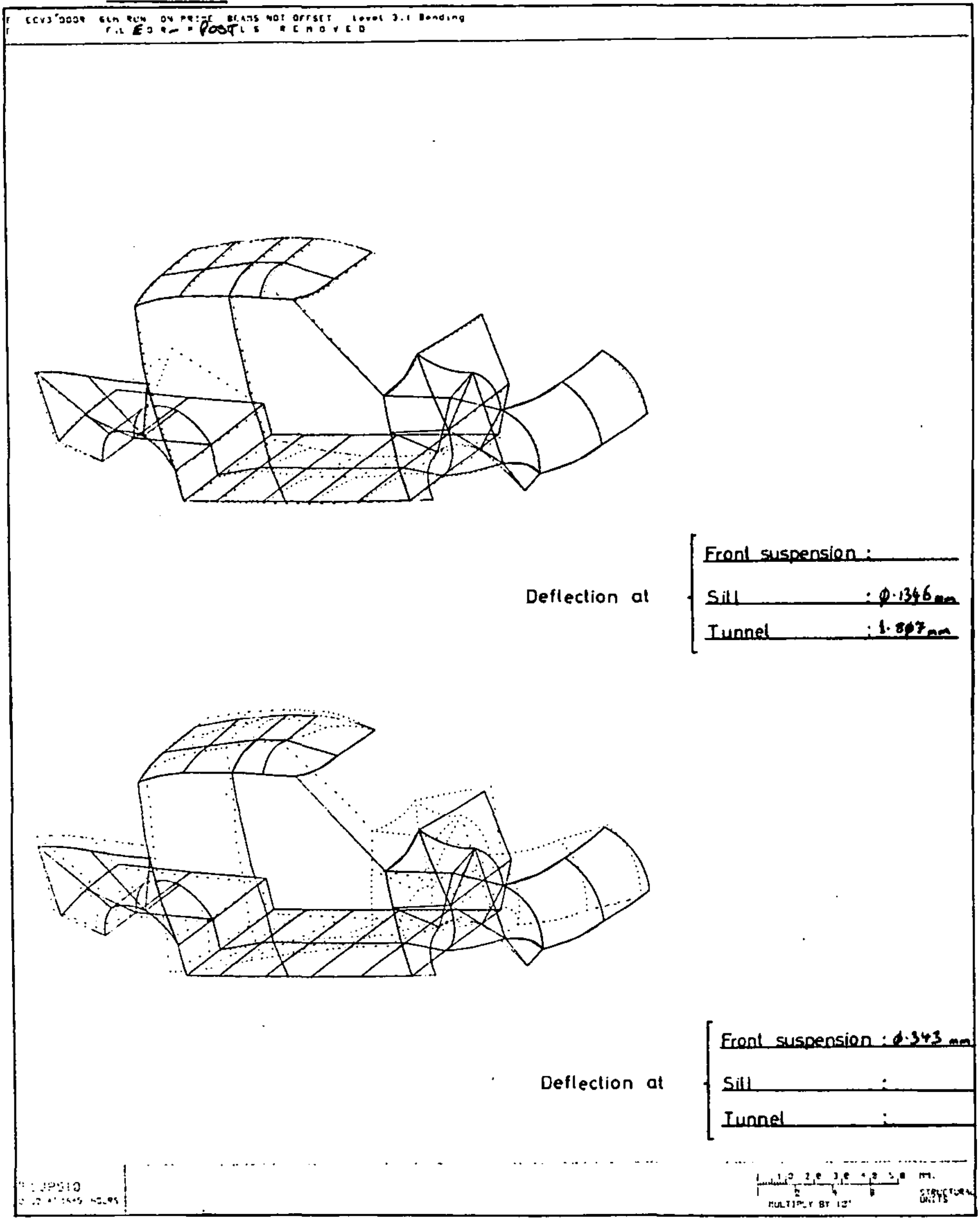
Figure 3.5 Roof removed - deflection plot.



Deflection at
Front suspension :
Sill : 0·1529 mm
Tunnel : 1·811 mm

Deflection at
Front suspension : 0·363 mm
Sill :
Tunnel :

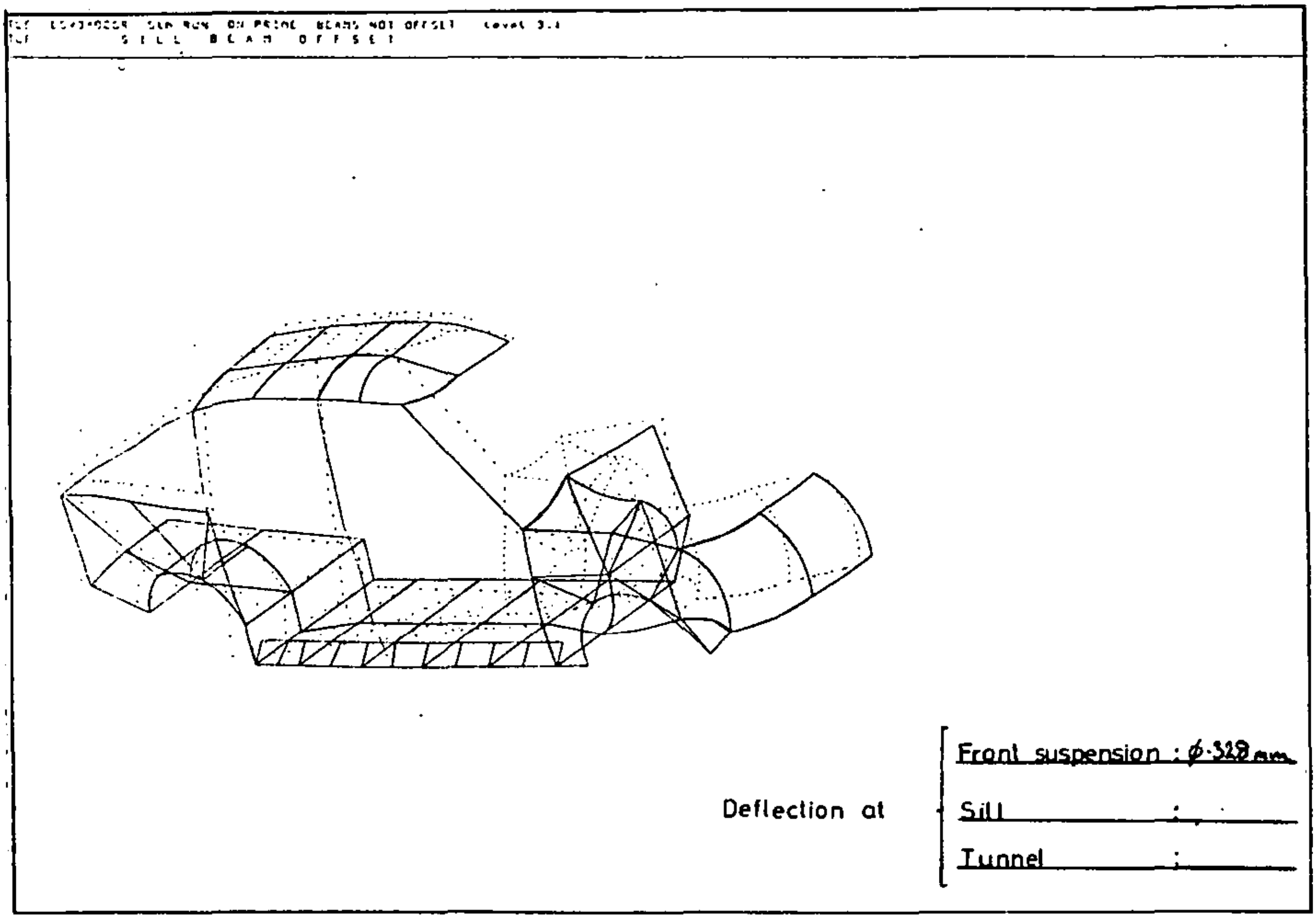
Figure 3.6 E-post removed - deflection plot.



Deflection at
Front suspension :
Sill : 0·1346 mm
Tunnel : 1·807 mm

Deflection at
Front suspension : 0·343 mm
Sill :
Tunnel :

Figure 3.7 Sill Offset



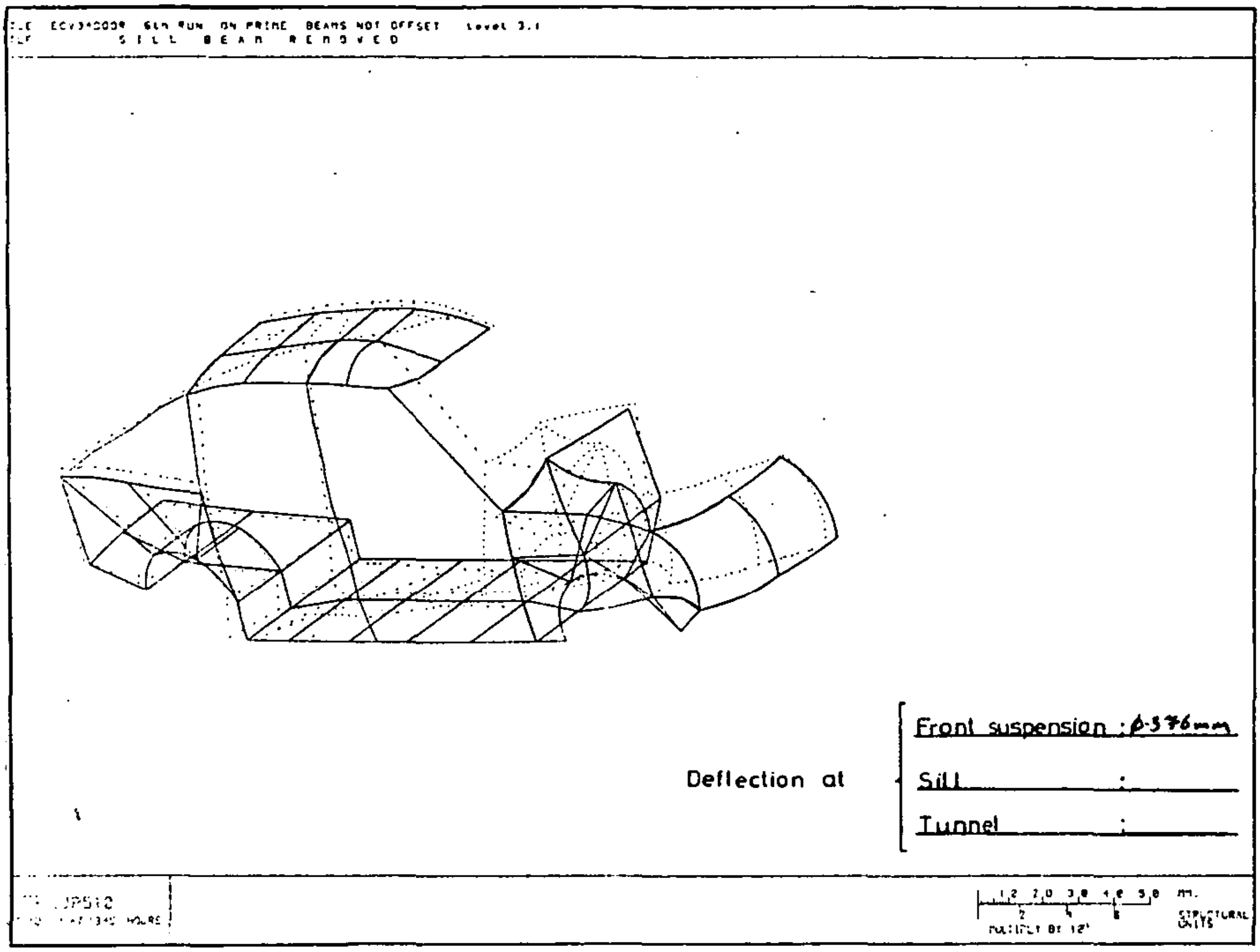
Deflection at

Front suspension : 0·328 mm

Sill :

Tunnel :

Figure 3.8 Sill removed - deflection plot.



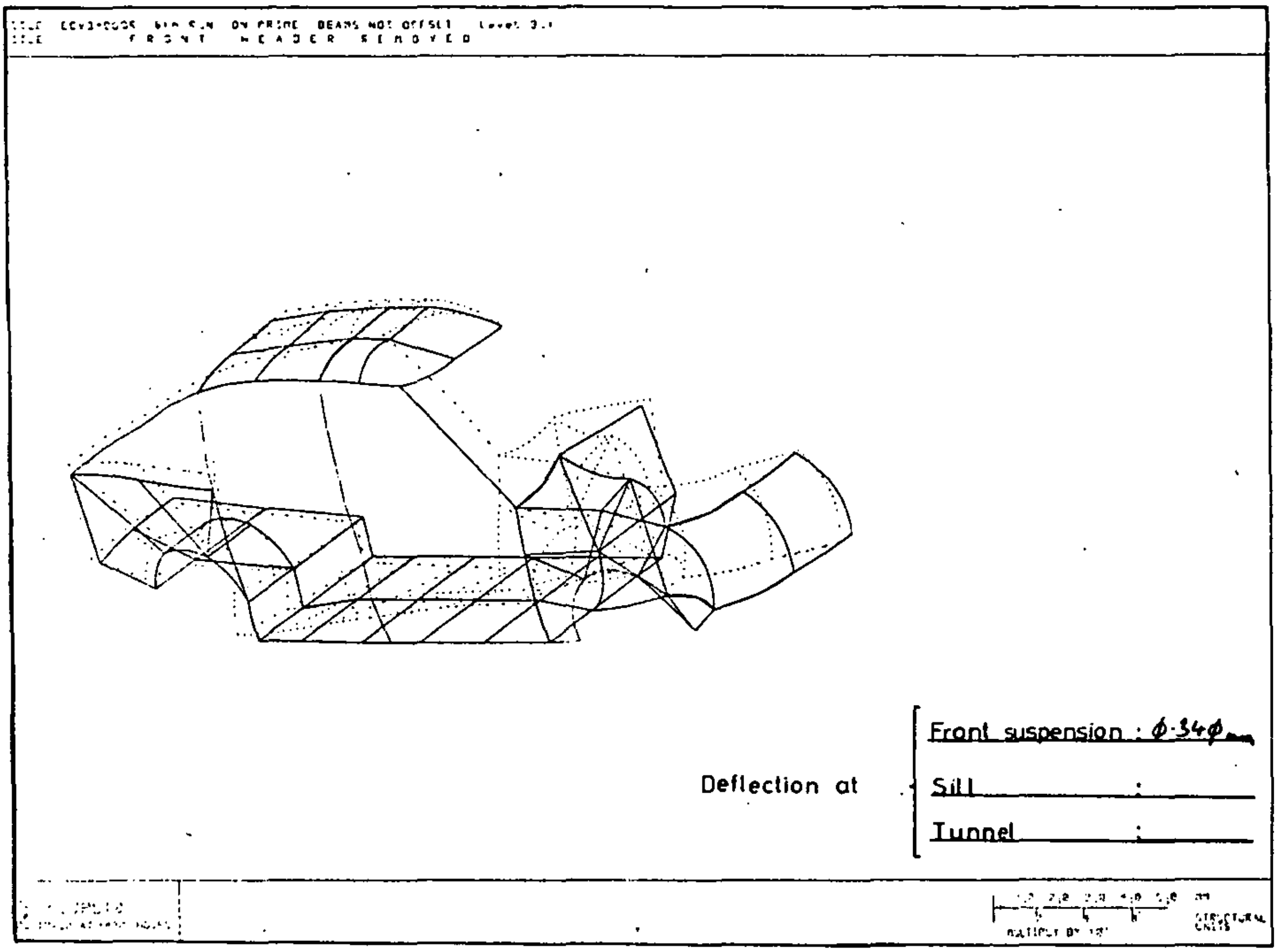
Deflection at

Front suspension : 0·376mm

Sill :

Tunnel :

Figure 3.9 Front Header removed - deflection plot.



Deflection at

Front suspension : 0·340mm

Sill :

Tunnel :

Figure 3.10 Tunnel removed - deflection plot.

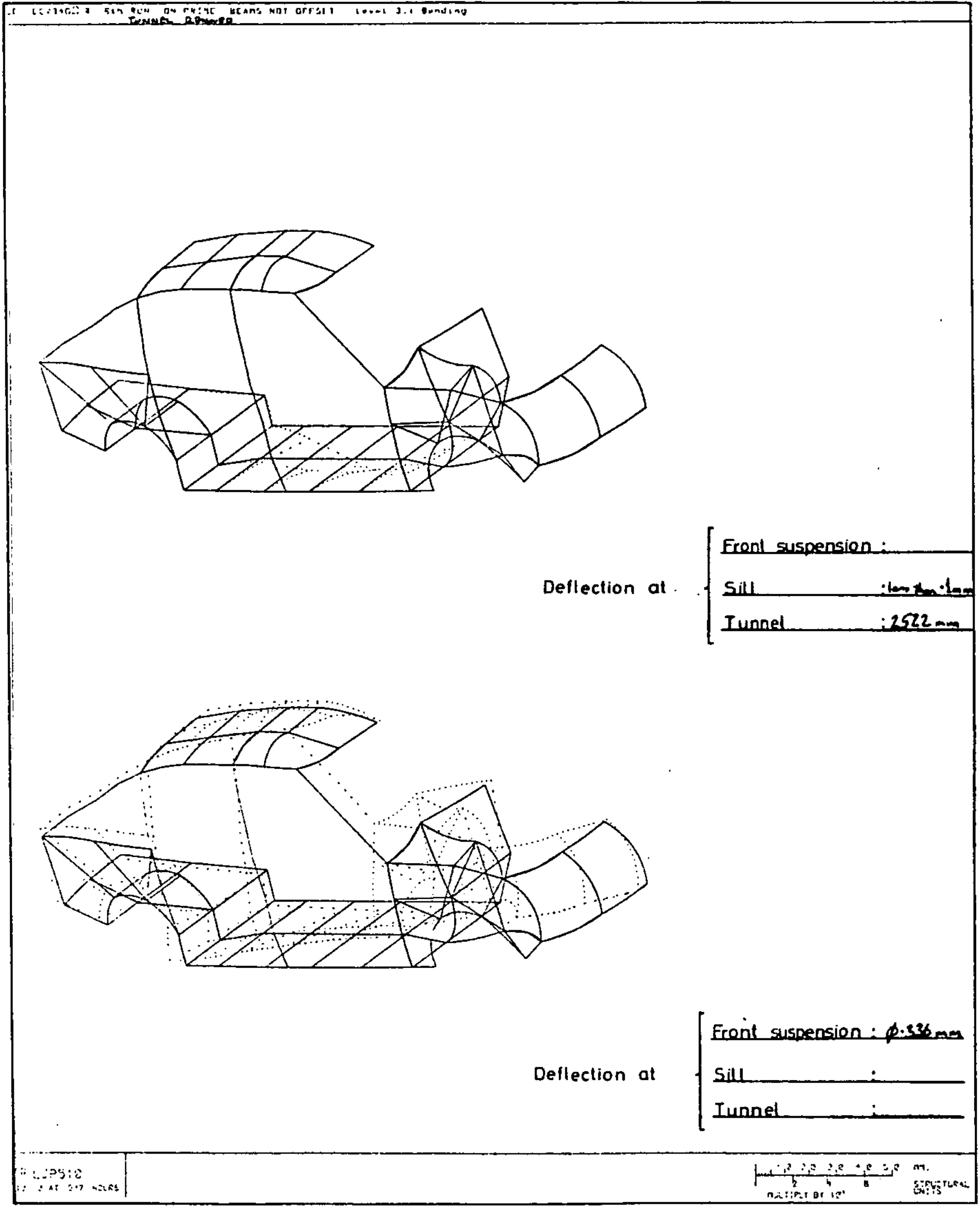
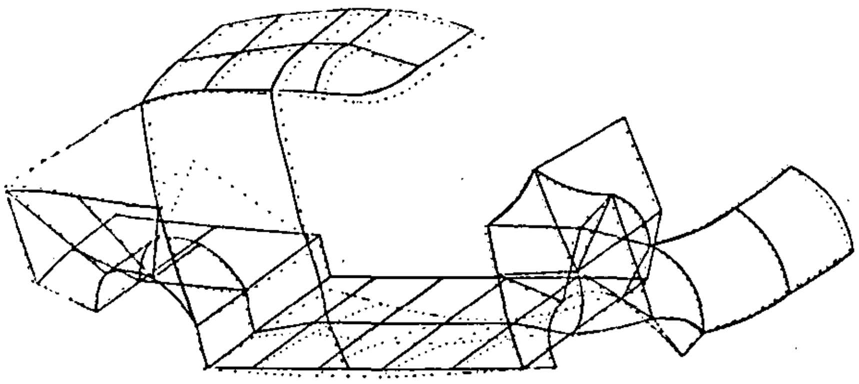
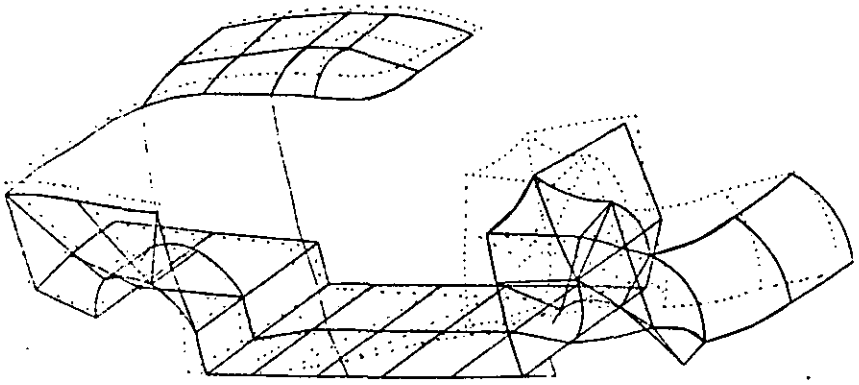


Figure 3.11 Windscreen pillar removed - deflection plot.

DCV3-3008 5th RUN ON PRIME BEAMS NOT OFFSET Level 3.1 Bending
WINDSCREEN PILLAR REMOVED

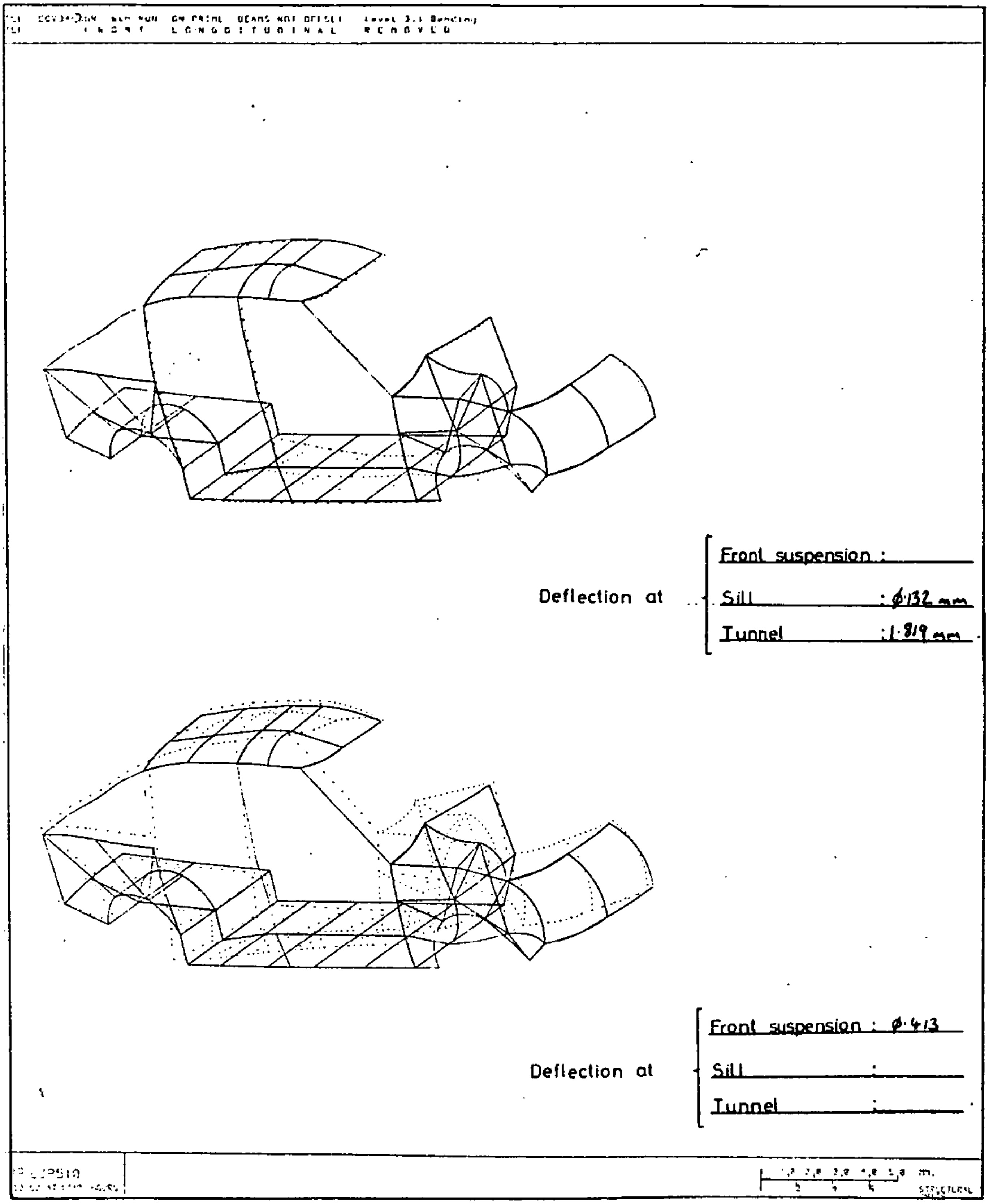


Deflection at
Front suspension :
Sill : 0·288 mm
Tunnel : 1·823 mm



Deflection at
Front suspension : 0·385 mm
Sill :
Tunnel :

Figure 3.12 Front longitudinal removed - deflection plot.


BEAMS NOT OFFSET Level 3.1 Bending
FRONT LONGITUDINAL REMOVED
m.
STRUCTURAL

Deflection at

Front suspension :

Sill : 0·132 mm

Tunnel : 1·819 mm

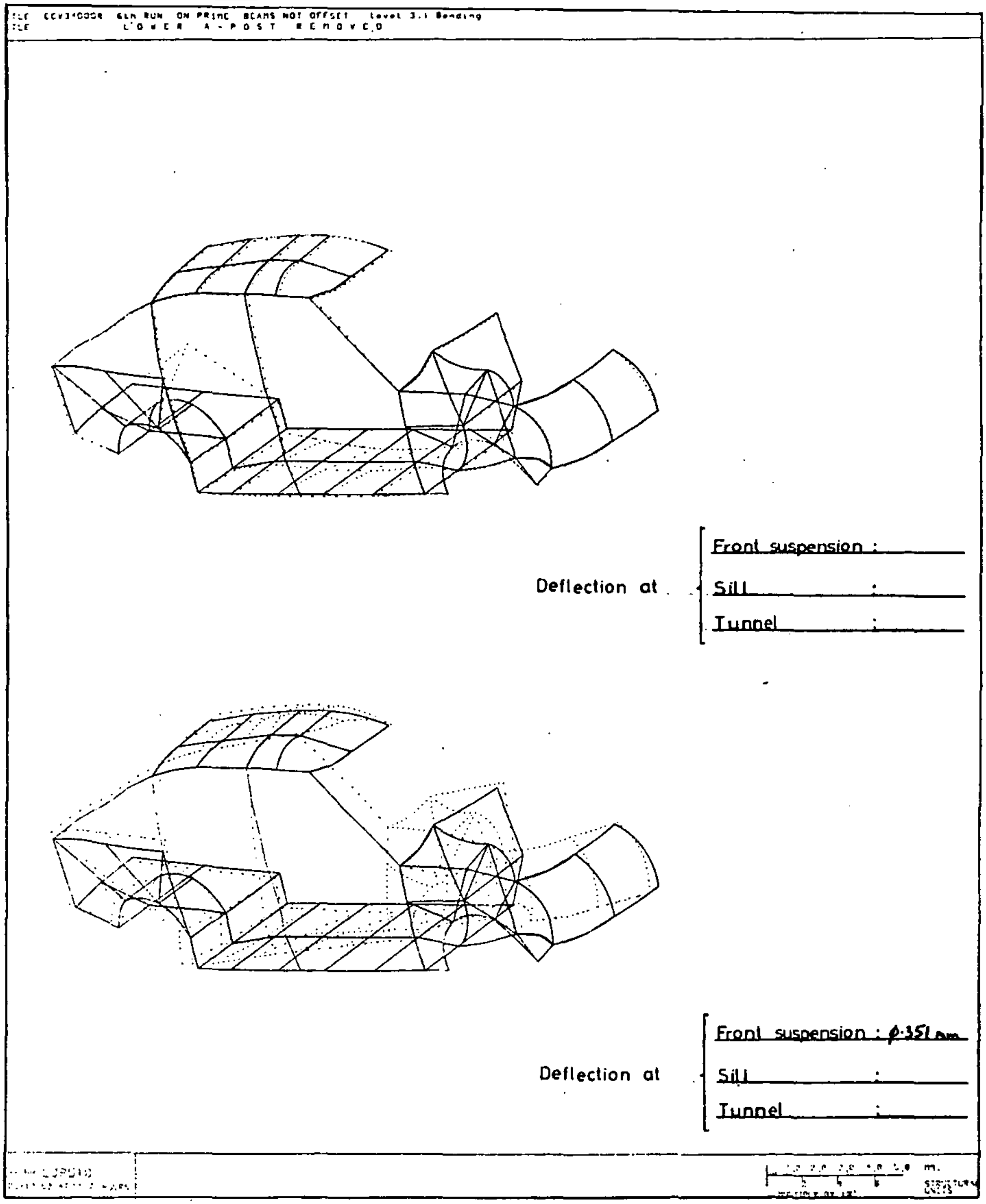
Deflection at

Front suspension : 0·413

Sill :

Tunnel :

Figure 3.13 Lower A-post removed - deflection plot.

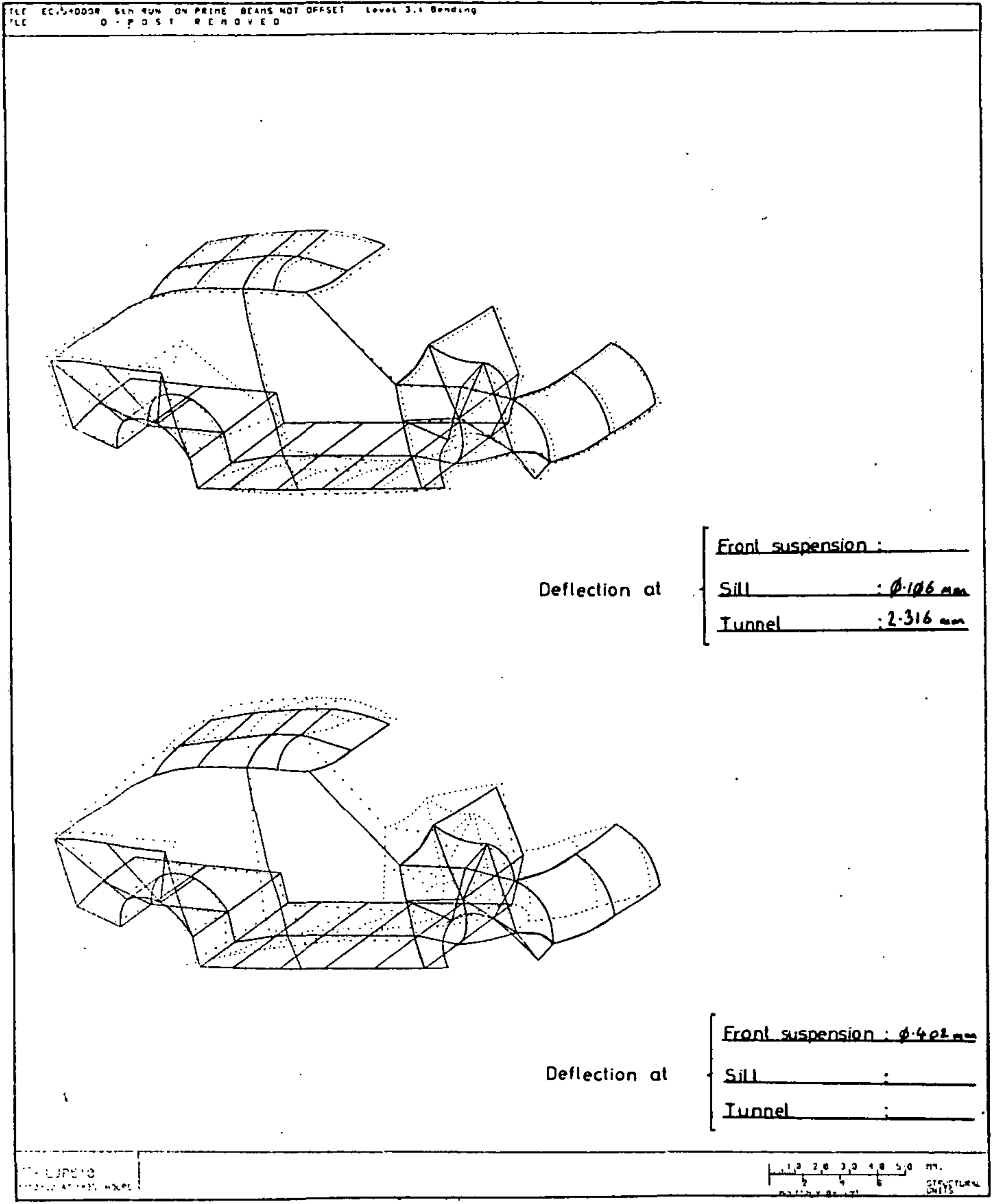


CCV3*COOCR GEN RUN ON PRIME BEAMS NOT OFFSET Level 3.1 Bending
LOWER A-POST REMOVED

Deflection at
Front suspension :
Sill :
Tunnel :

Deflection at
Front suspension : 0·351 mm
Sill :
Tunnel :

Figure 3.14 D-post removed - deflection plot.

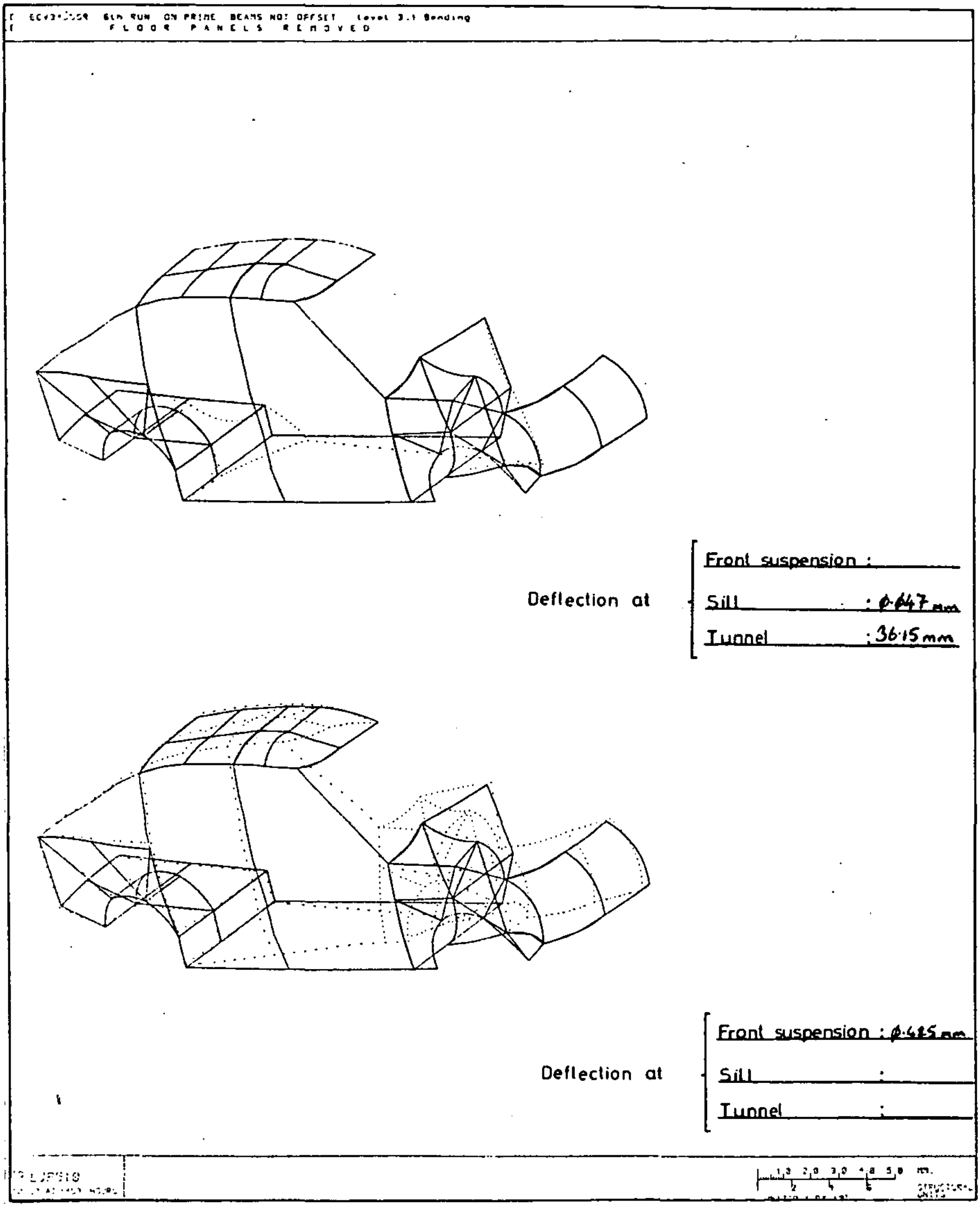


TLE EC.54DOOR 5th RUN ON PRIME BEAMS NOT OFFSET Level 3.1 Bending
TLE D-POST REMOVED

Deflection at
Front suspension :
Sill : 0·106 mm
Tunnel : 2·316 mm

Deflection at
Front suspension : 0·402 mm
Sill :
Tunnel :

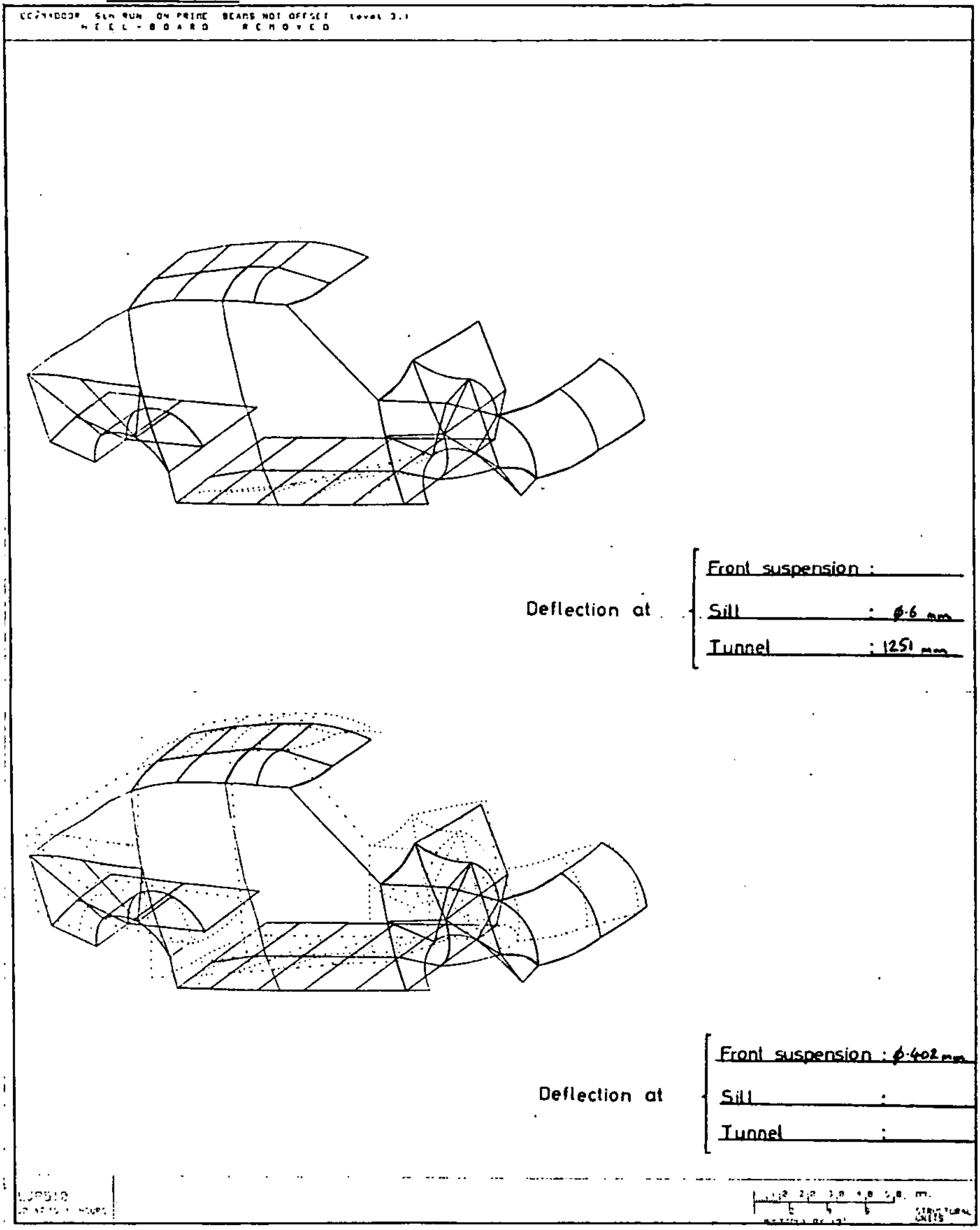
Figure 3.15 Floor panels removed - deflection plot.



Deflection at
Front suspension :
Sill : 0·647 mm
Tunnel : 36·15 mm

Deflection at
Front suspension : 0·425 mm
Sill :
Tunnel :

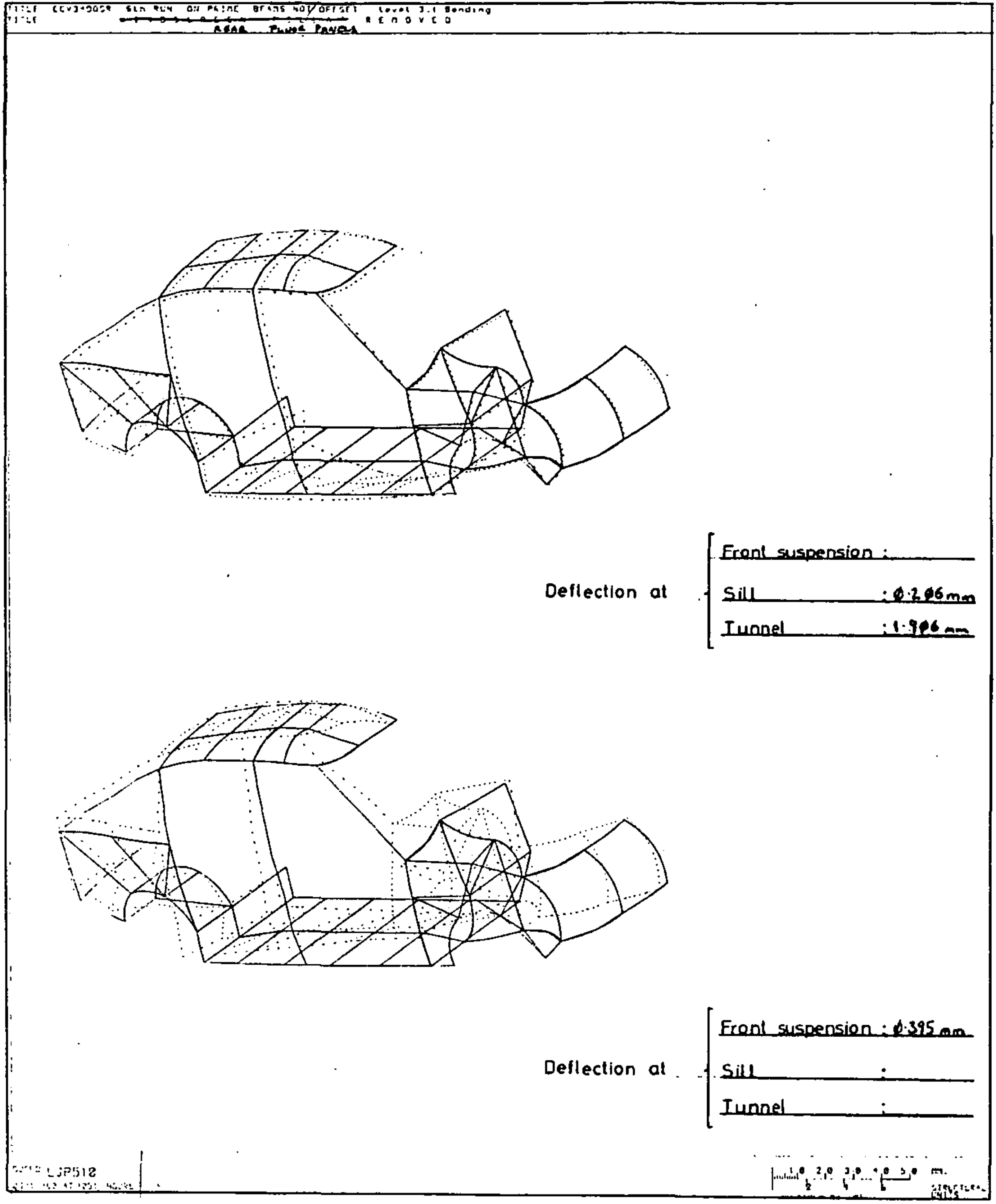
Figure 3.16 Heel board removed - deflection plot.



Deflection at
Front suspension :
Sill : 0.6 mm
Tunnel : 1251 mm

Deflection at
Front suspension : 0.402 mm
Sill :
Tunnel :

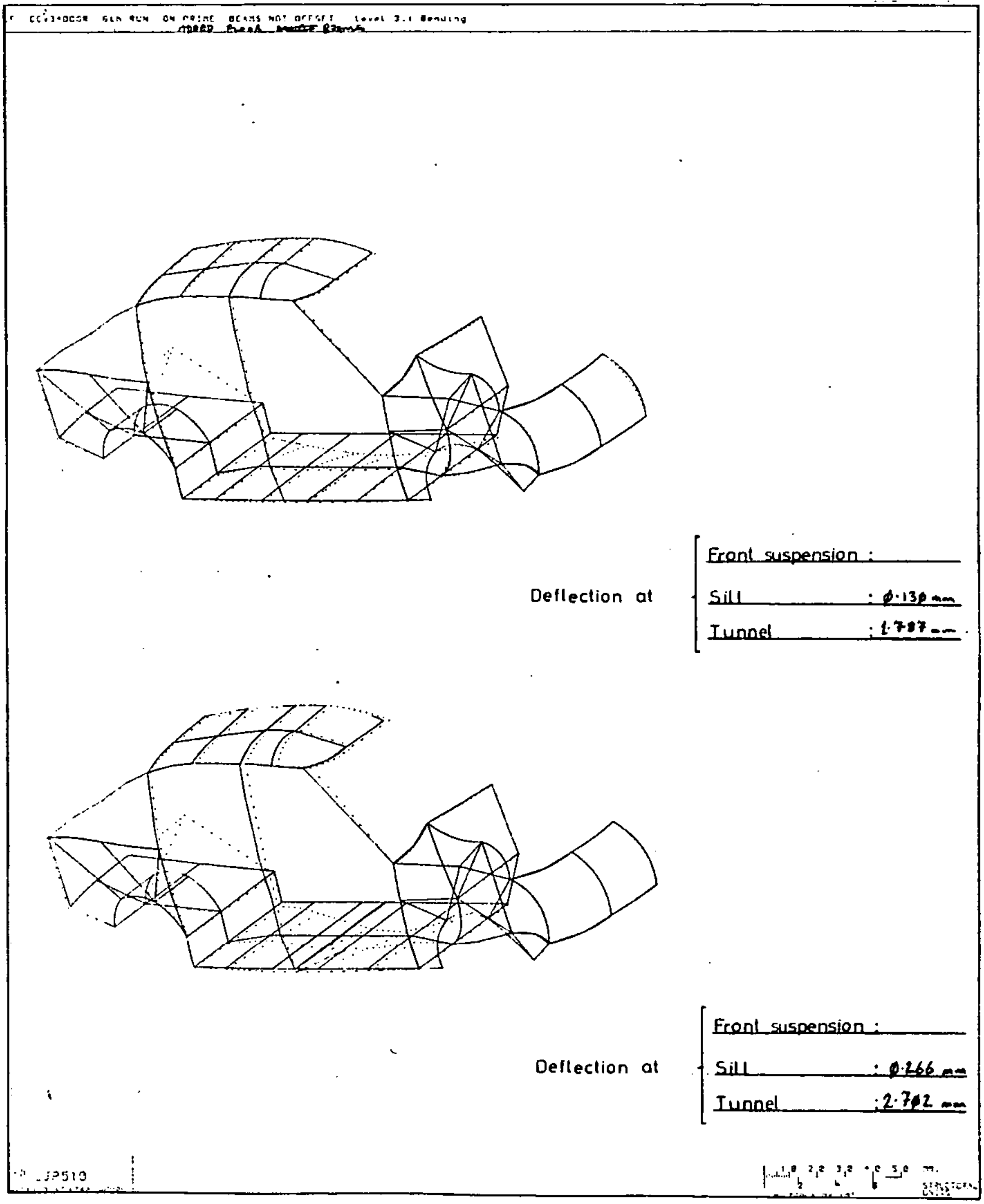
Figure 3.17 Rear floor removed - deflection plot.



Deflection at
Front suspension :
Sill : 0·206 mm
Tunnel : 1·906 mm

Deflection at
Front suspension : 0·395 mm
Sill :
Tunnel :

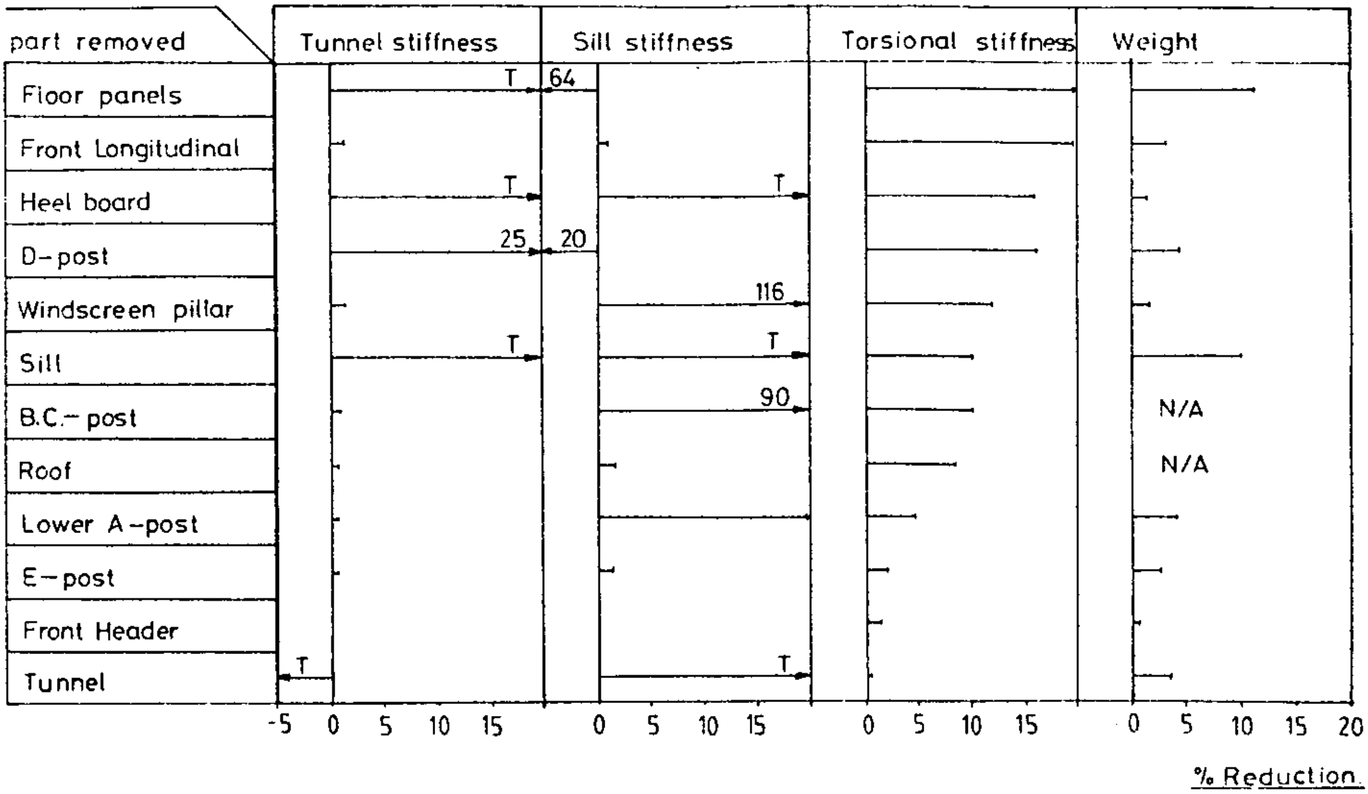
Figure 3.18 Cross beams added to floor.



Deflection at
Front suspension :
Sill : 0.130 mm
Tunnel : 1.787 mm

Deflection at
Front suspension :
Sill : 0.266 mm
Tunnel : 2.702 mm

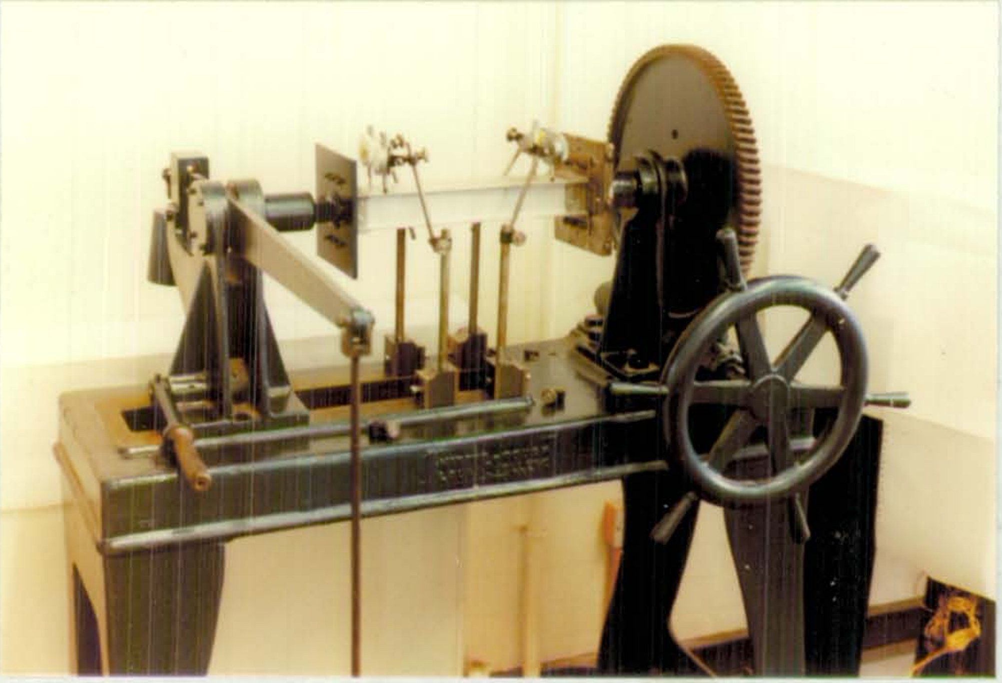
Figure 3.19 Histogram of Stiffness and Weight Changes.



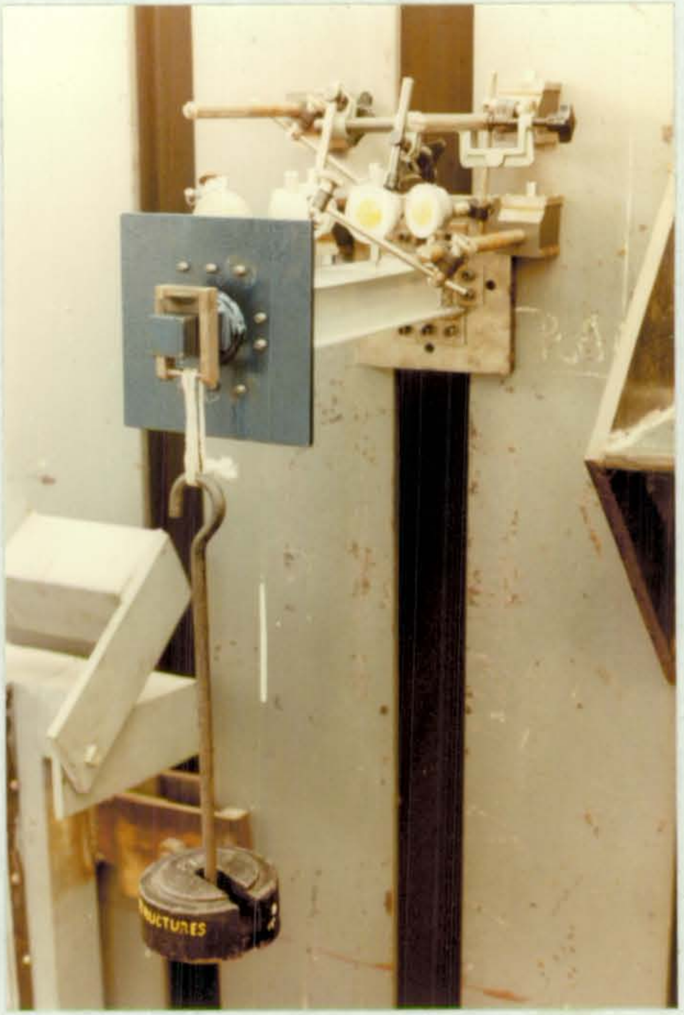
T indicates total failure

PLATE 1

Beam Stiffness Measurements



(a) Torsional Stiffness Rig



(b) Bending Stiffness Rig

The experimental measurement of beam stiffnesses.

This work was undertaken to assess the ability of simple theory to account for the different fabrication techniques which may be employed in future vehicles. It is common practice to use programs based on simple beam theory and the Batho-shear equations to calculate the properties of complex thin wall beam cross sections. This, however, assumes that the beam is a continuous construction and therefore tends to produce an over optimistic set of properties. To take account of different fabrication techniques is quite a complicated task, and requires some basic experimental work as a guide line to the value of the analytical results obtained.

Experimental investigations of this kind have been published previously by Sharman (3), Myers , but these were confined to spotwelded beam sections. Both authors obtained experimental stiffnesses of between 75% and 85% of simple theoretical predictions. It came as somewhat of a suprise then, when two papers were published (11,12) claiming a doubling in stiffness when adhesives are used instead of spotwelds, since it was apparent from previous work that there was not that much efficiency to be regained. As a result of this it was decided to repeat the experiment as described in references (11,12) to investigate their findings.

4.1 Presentation of results by previous workers.

The results presented by Sharman and Myers are shown in table 4.1. The work by Sharman was an investigation into the stiffness changes induced by a decrease in the number of spotwelds used. It can be seen that at no time does the efficiency fall below 80% and, with a pitch of 25mm, the torsional stiffness is also around 80%. Sharman states that yielding occurred in the spotwelds when using a large pitch while trying to induce measurable deflections and it is clear that Myers had similar problems as his results are non-linear and yield a torsional efficiency of only 65% with 50mm spotweld pitch.

The results produced in references (11,12) are not presented in the same manner as those in (3) and no calculations of theoretical stiffnesses are presented by the author. Taking the sizes of the cross-section to be those shown in Figure 4.1 (from ref. 12) it is possible to calculate some approximate theoretical stiffnesses (they will not be exact as these are only nominal dimensions). It is now possible to produce a set of 'efficiencies' from the results (see table 4.2). Here some discrepancies seem to appear. The measured stiffness for the adhesive bonded aluminium beams is considerably higher than that which would be predicted by simple theory. It can be demonstrated that a 10% discrepancy in one of the beam dimensions can result in a 15% change in torsion constant and second moment of area, but even this is not enough to explain efficiency values of 150%. A closer look at the raw results, presented to the sponsoring company, shows that in many cases a similar pair of beams were tested. These results often show one measurement to be of an expected magnitude, while the second stiffness may be twice as high. To obtain the quoted results the researcher had simply taken the average of the two widely varying results to obtain the quoted Figures. The value of these results is not very definite since no corrections are made for the actual beam sizes, thus allowing manufacturing variations to swamp any trend which may be present due to the fabrication technique. It is also a cause for concern that for the two riveted beam sections, one steel and one aluminium, the aluminium section was considerably stiffer than the steel, whereas it would be expected that the steel section should be approximately three times stiffer than the aluminium.

When presented to the sponsoring company these results caused considerable confusion. Accordingly it was decided that the experiments should be repeated under more carefully controlled conditions.

4.2 Experimental Technique and Specimens.

The beams used by this author were somewhat smaller in cross-section than those used by the other authors to allow greater deflections to occur without the risk of overstressing. All the previous workers had used beams with cross-sectional properties similar to those of a fairly large sill section. These are naturally extremely stiff and require large loads to yield easily measureable deflections. This induces problems such as support deflections and the possibility of small amounts of local buckling on the beam around the supports, both of which could be significant with respect to the small deflections due to the bending of the beam. With this in mind it was decided, in this investigation, to use a beam size more akin to a cantrail section. The shape was kept the same as that used by previous workers to maintain comparability between the results.

The beams were fabricated from either steel or aluminium and were connected using rivets, spotwelds or adhesive (dimensions are given in Figure 4.2). The tests were developed to measure both the torsional and bending stiffnesses of the beams. The torsion tests were done in a torsion machine and the bending tests were done using the beam as a cantilever on a structural test bed.

4.2.1 The bending test.

The deflection at any distance x along a cantilever beam is given by the equation: (neglecting shear deformation)

$$\boxed{d_y = \frac{w\,(Lx^2 - x^3/3)}{2EI}}$$

To allow for possible shear and rotational deflections at the support it is necessary to take three measurements of deflection at different distances along the beam. It is then possible to separate the support deflections and be left with just the beam deflection. If the start of the beam is assumed to be at the measured point nearest to the root (1) we can then say that the deflections at the remaining points (2) and (3) are in effect only:

$$d_{2a} = d_2 - d_1$$

and

$$d_{3a} = d_3 - d_1$$

if a rotation of θ is also assummed to occur at (1), then:

$$d_{2a} = (d_2 - d_1) - \theta\ x_2 \tag{1a}$$

and

$$d_{3a} = (d_3 - d_1) - \theta\ x_3 \tag{1b}$$

now

$$d_{2a} = \frac{w(Lx_2^2 - x_2^3/3)}{2EI} \tag{2a}$$

and

$$d_{3a} = \frac{w(L_3^2 - X_3^3/3)}{2EI} \tag{2b}$$

Substituting 1a into 2a gives:

$$\theta = -(\frac{w}{2EI}\ (Lx_2^2 - x_2^3/3) - (d_2 - d_1))/x_2 \tag{3}$$

Substituting 3 into 2b gives:

$$(d_2 - d_1) - \frac{x_3}{x_2}\ (d_2 - d_1) = \frac{w}{2EI}\ (Lx_3^2 - x_3^3/3) - \frac{x_3}{x_2}\ (Lx_2^2 - x_2^3/3) \tag{4}$$

Which gives:

$$\boxed{EI = \frac{w}{2}\,\frac{L(x_3^2-x_3^3/3) - \frac{x_3}{x_2}(Lx_2^2-x_2^3/3)}{(d_3 - \frac{x_3}{x_2}d_2 + (\frac{x_3}{x_2}-1)d_1)} \qquad 5}$$

Unfortunately, two nearly equal quantities are being subtracted in the denominator. This can result in large errors in calculated EI values from fairly small errors in the deflection measurements. It is possible for a data error of 2% to result in a 20% error in EI, the likely error becoming larger with an increased beam stiffness. This means that it is imperative to exercise extreme care in the measurement of these deflections. The experiment must be repeated many times and a statistical record made of the results to ensure that an adequate level of precision is achieved. This was assumed to be the case with a correlation coefficient of better than 0.999.

The layout of the experiment was as shown in Figure 4.3. The root end plate was securely bolted to a sturdy upright structural test rig. The plate at the free end carried the load and was securely bolted to the flanges on the beam. Six dial gauges were used in three pairs one of each on either side of the beam, and each pair equally spaced along the beam, but at least 50mm from the ends (to allow the end effects to diffuse out). The positions of the gauges were measured carefully by the dimensions X2,X3 and L. The gauges had to be positioned very carefully in the vertical direction to avoid erroneous readings due to sliding contact. Some of the beams were distorted slightly during manufacture which exacerbated this problem.

To obtain a statistically reliable result a series of ten loading cycles were carried out. It was noticed that there was some settling-in during the first 2 or 3 loading cycles, during which the measured value of EI dropped gradually. Since this settled out it was concluded that there was some initial permanent movement at the support bolts. The results obtained are shown in table 4.1.

4.2.2 The torsion test.

The torsion rig (Figure 4.4) consists of a rigid support for a square shaft at one end, and a bearing mounted support at the other , which is fixed to a torque arm. The beam sections were fitted into the rig by affixing end plates with square pegs on to the end flanges of the beam.

To avoid any end effects an initial test was carried out to find the distribution of the rate of twist along the beam. This showed that the end fixings had no effect at any distance greater than approximately 50mm from the beam ends. For the best accuracy of results the maximum possible gauge length had to be used. This required the dial gauges to be placed at a distance of 50mm from the end and 7mm in from the edge of the beam. In this way it is possible to obtain large deflection readings without applying excessive loads (see Figure 4.4). The total twist over the gauge length is given by:

$$\boxed{\dot{\theta} = d_1 - d_2 - d_3 + d_4/aL}$$

and the torsional stiffness is given by:

$$\boxed{K = GJ = T / \dot{\theta}}$$

The readings taken were repeatable, and the loadings were kept low and caused no permanent set. One adhesive bonded beam was observed to suffer from creep. This was assumed to be caused by a poorly prepared adhesive layer. The results for this beam are not included with the rest in Table 4.1.

4.3 Discussion of results - Torsion.

4.3.1 Aluminium -Rivets (Sample A1)

The riveted section had by far the worst efficiency. This was probably due to stress concentrations around the rivets causing a reduction in the effective plate stiffness Et. Also any lack of fit will allow weakening local distortions to occur. These problems were not pursued further as the results had only been included for completeness in the comparison with other workers. Rivets are far from ideal for use in motor vehicles as they loosen with vibration of the structure.

4.3.2 Aluminium - Spotwelds (Samples A2 and A3)

The spotwelds show a marked improvement over the riveted section, both beams giving an efficiency of 86-87%. These results were reassuringly consistent, but were slightly higher than those found by Sharman or Myers. There may be some size effect which causes a joint to have a greater effect on a larger beam. Also, using smaller section beams results in higher deflections that can be measured more accurately. Stresses were also lower giving less possibility of yielding occuring in the spotwelds.

4.3.3 Aluminium - Adhesive (Samples A4,A5,A6 and A7)

Here the efficiency improved again, now up to 93% of theory. The actual stiffness increase is around 40% since the shear flow path is different from the previous two cases, reducing the effective length of the periphery (ds), see Figure 4.5. The basis for the efficiency quoted is a homogeneous beam section and no attempt was made to account for the adhesive layer. To account for the adhesive requires a complex analysis using St. Venant's Torsion theory, see Chapter 5.

4.3.4 Steel - Spotwelds (Sample S2)

The efficiency of the steel beam was almost identical to that measured for the aluminium beam, which is to be expected if the spotwelds are of equal quality.

4.3.5 Steel - Adhesive (Samples S3 and S4)

Even with the high ratio of modulus of rigidity between the steel and adhesive, there is very little loss in efficiency. The steel specimen is unexpectedly more efficient than the aluminium one, this can be explained since the adhesive layer in the steel sample is so thin that it has little flexibility and thus does not weaken the beam.

Bending.

4.3.6 Aluminium - Rivets

The efficiency at 80% was again very much lower than the other beams. Here the same comments apply as for the torsion case. Lack of fit of the rivets stops the loads from being transferred effectively out to the faces of the beam.

4.3.7 Aluminium - Spotwelds

The efficiency of these beams was almost perfect (98%) showing that spotwelds achieve a good load transfer to the top and bottom plates. This result is much higher than that obtained by Sharman or Myers.

4.3.8 Aluminium - Adhesive

This beam exhibited an almost identical stiffness to the spotwelded specimen, there being almost no efficiency loss. This is predicted by a simple analysis shown in Chapter 5.

4.3.9 Steel - Spotwelds

This result is again almost perfect, but unexpectedly is slightly lower than the aluminium beam, in which the spotwelds would not be expected to be as reliable, aluminium being difficult to spotweld. The results for the steel beam will not be as accurate as those for aluminium as the beam is stiffer resulting in smaller deflections and a greater possibility of local distortions.

4.3.10 Steel - Adhesive

The results are again as expected with a very small drop in efficiency from the theoretical value. With such a thin adhesive layer there is probably a severe stress concentration at the root of the joint.

4.4 Photo-elastic study of fabricated beam stresses.

These experiments were carried out to compare with the analytical results obtained in chapter 5. The stresses were measured on the closing plates of the beams used for the stiffness measurements in this chapter. It was specifically of interest to find the reason for the torsional efficiency drop found with the spotwelded beams. All the tests were carriedout in torsion since this was the mode which showed the greatest drop off in efficiency. Three beams were tested, they were: a) steel/-adhesive, b) aluminium/adhesive and c) aluminium/spotweld.

4.4.1 Procedure for test.

A 0.121" (3.07mm) thick plate of birefringent epoxy resin was adhered to the surface of the beam samples in the usual manner and the beams were fitted into a torsion testing machine. The torque was applied to the beam so that between 2 and 3 fringes appeared on the photo-elastic coating when viewed through the reflection polariscope.

Stresses were measured using a compensator with a correction factor of 1/49. From Figure 4.7, with an epoxy thickness of 0.121" and a strain-optical coefficient (K) of 0.15 we get a value of 618$\mu\xi$/fringe for the coating.

4.4.2 Sample calculation.

For the aluminium/spotweld beam, from Figure 4.6, the maximum compensator reading (over the centre of the closing plate) is 150. This gives a maximum fringe order of 150/49=3.061 which gives a maximum strain of 3.061x618=1892$\mu\xi$.

$$\boxed{\begin{aligned}
&\varepsilon_1-\varepsilon_2=1892\times10^{-6}=\frac{1+\upsilon_s}{E_s}(\sigma_1-\sigma_2)_s\\
&\sigma_1-\sigma_2=101\times10^{6}\ \mathrm{N/m^2}\\
&\tau=\tfrac{1}{2}(\sigma_1-\sigma_2)\sin 2\theta=\underline{50.5\times10^{6}\ \mathrm{N/m^2}}
\end{aligned}}$$

This value may be compared with the batho-shear predicted stress of 34.2×10^6 under the same load of 112.6Nm. This shows that the actual peak stress on the beam surface is about 50% higher than expected. However, Figure 4.6 shows that the stress is by no means constant, varying over the surface from 35 to $50 N/m^2$ with the peak stresses occurring between the spotwelds. This birefringent method is very susceptible to large errors if bending is induced in the plates since they are exaggerated by the thickness ratio between the metal and the epoxy coating.

In a similar manner to this the maximum stress from the aluminium/adhesive beam was found to be 44.4×10^6 N/m^2 spread evenly over the centre of the top plate. This compares closely with the expected value of 40×10^6 N/m^2.

4.5 Conclusions.

4.5.1 An adhesive beam is substantially stiffer than a spotwelded beam in torsion for two reasons. a) The efficiency is improved by the more effective connection and b) the torsion constant is increased by redirecting the shear flow to give a shorter periphery.

A stiffness increase of between 8% and 10% may be expected due to the improved connection and a further 40% from the change in the shear flow path. Overall an improvement of 50% in torsional stiffness should be produced by a conversion from spotwelds to adhesive.

4.5.2 In bending, the efficiency of the joining method does not seem to be so vital as it is in torsion. Both spotwelds and adhesives give results close to simple theory. Only the very ineffective rivets show a substantial decrease in stiffness from theoretical prediction.

The results show no increase in bending stiffness due to the improved potential to transfer loads to the closing plates by using adhesives in place of spowelds.

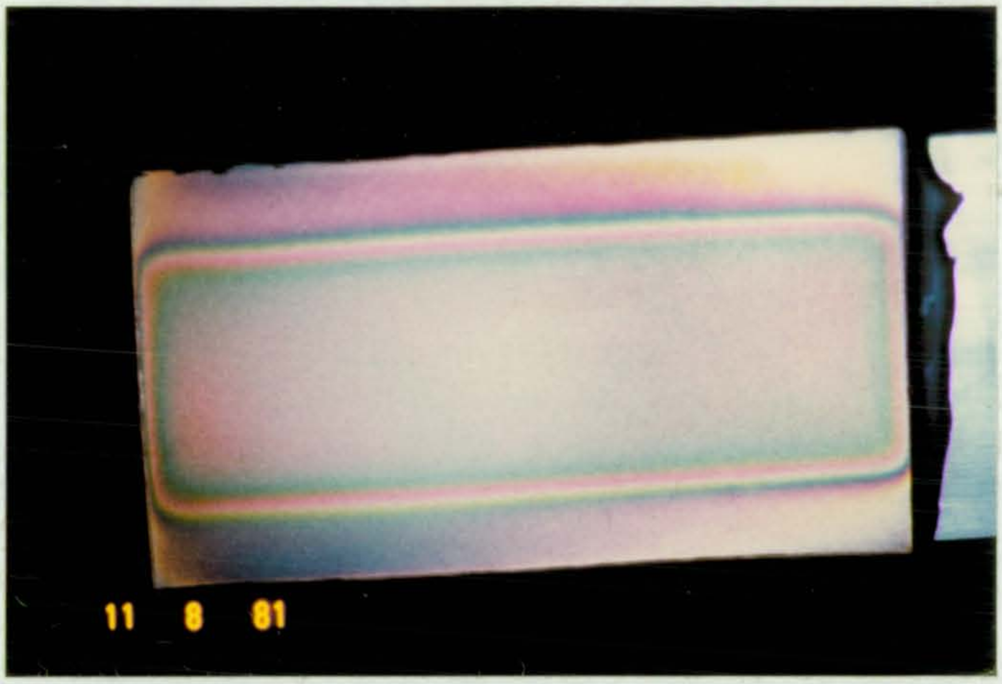
4.5.3 Inspecting the results of previous workers, it appears that larger beams are comparatively more affected by fabricated joints than the smaller beams tested here. Doubling the dimensions of the beam to those used by Sharman gives a 5% reduction in efficiency.

4.5.4 The photo-stress results show a much more even stress distribution for the adhesive bonding than for the spotwelds. The increase in stress on the spotwelded beam accounts for the loss in efficiency of this beam .

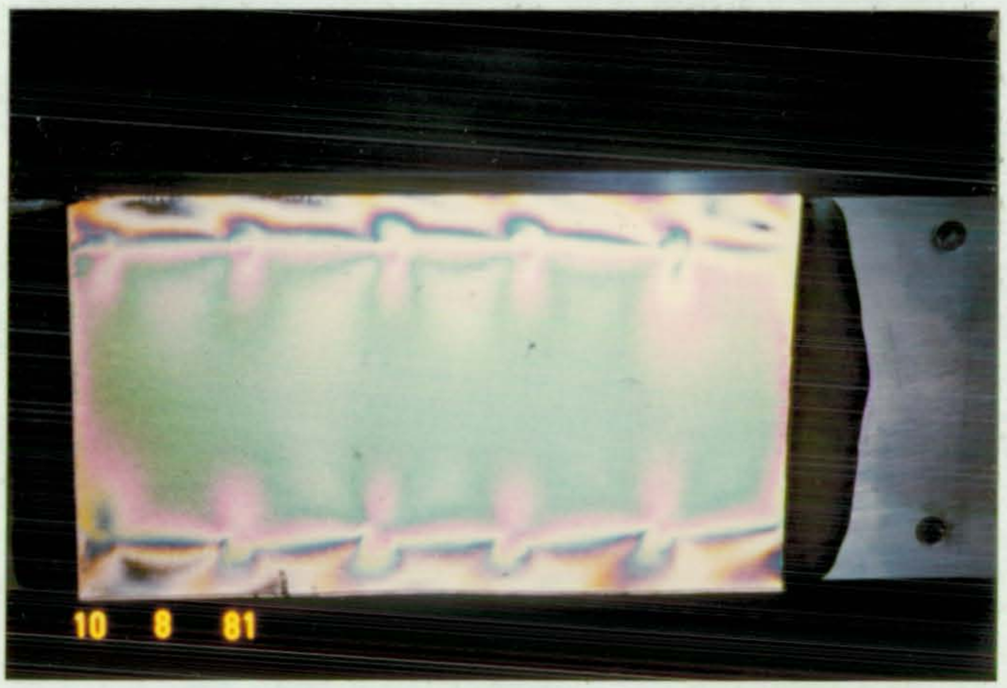
The adhesive bonded beam shows the expected pattern of stress as predicted in chapter 5, with a sharp stress concentration at the joint root and very little stress in the flange itself.

PLATE 2

Photostress Results On Beams



(a) Adhesive



(b) Spotweld

Table 4.1

Beam efficiency results

Beam	theoretical stiffness	measured efficiency
TORSION.		
Aluminium		
Rivets	1234Nm/rad	69%
Spotwelds	1234Nm/rad	86%
Adhesive	1696Nm/rad	93%
Steel		
Spotwelds	3771Nm/rad	87%
Adhesive	5184Nm/rad	96%
BENDING.		
Aluminium		
Rivets	18.97KN/m	80%
Spotwelds	18.97KN/m	98%
Adhesive	18.97KN/m	96%
Steel		
Spotwelds	55.85KN/m	93%
Adhesive	55.85KN/m	94%

Bending efficiencies are quoted to within 5%, and torsion to within 2%.

Table 4.2

Results presented by Sharman and by Myers

Spotweld Pitch (mm)	12.5	24	33	50	
Bending efficiency	89%	89%	85%	82%	
Torsional efficiency	81%	81%	-	-	Sharman
Torsional efficiency	-	77.5%	-	65%	Myers

Figure 4.1 Oxford Polytechnic beam cross section.

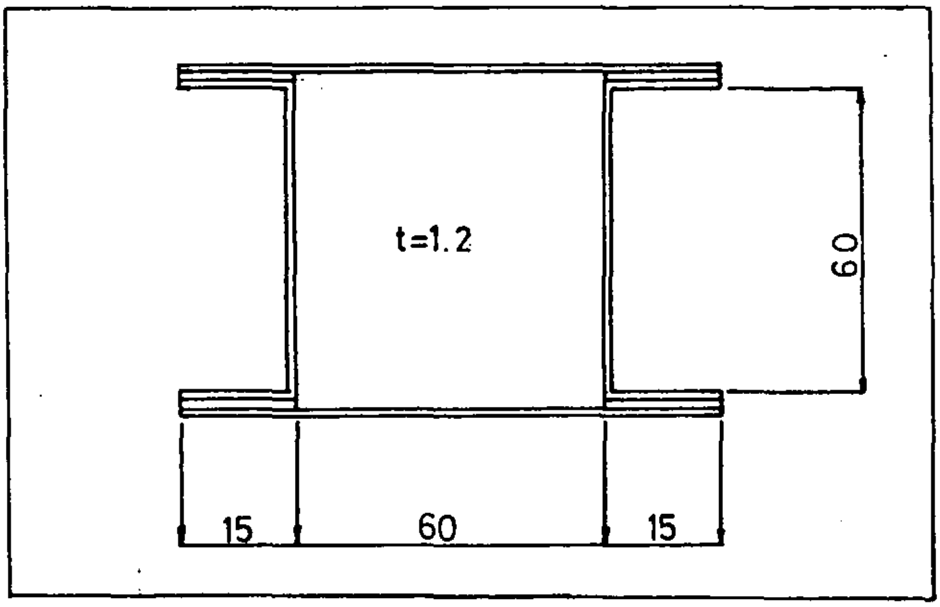


Figure 4.2 Experimental beam cross section (Loughborough).

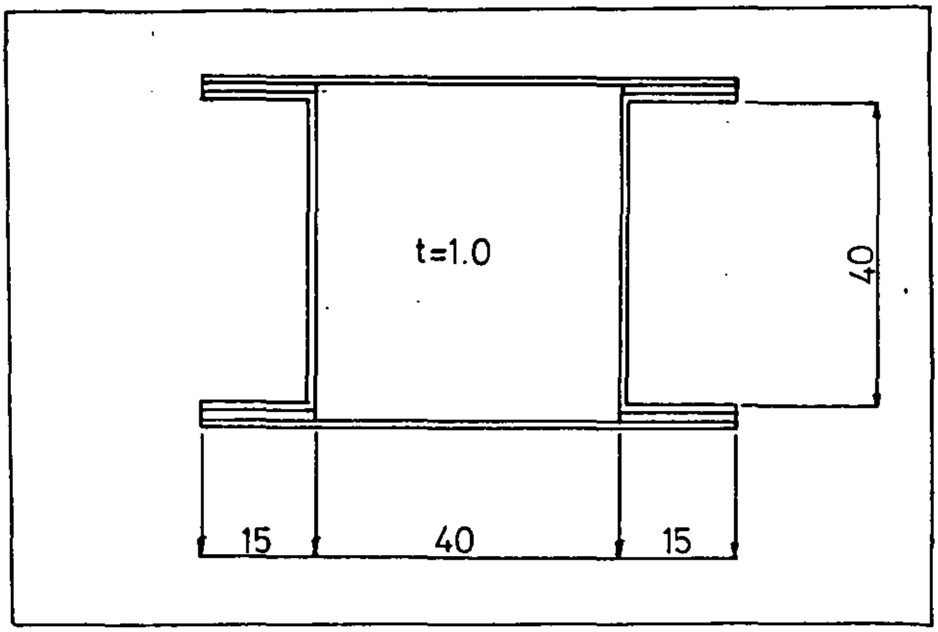


Figure 4.3 Bending experiment layout.

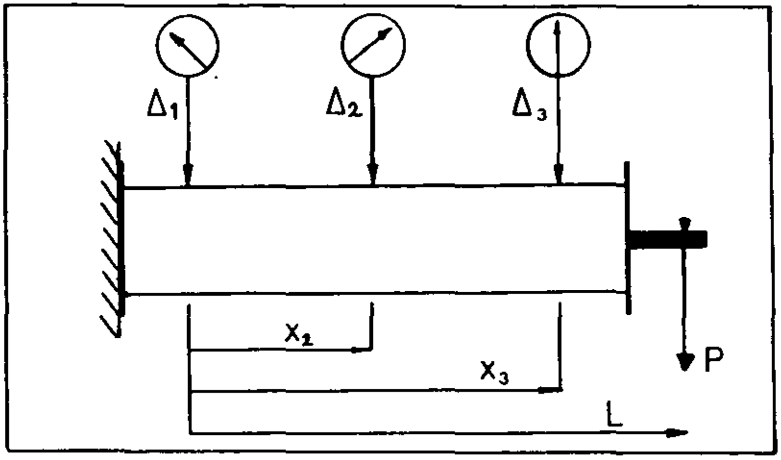


Figure 4.4 Torsion experiment layout.

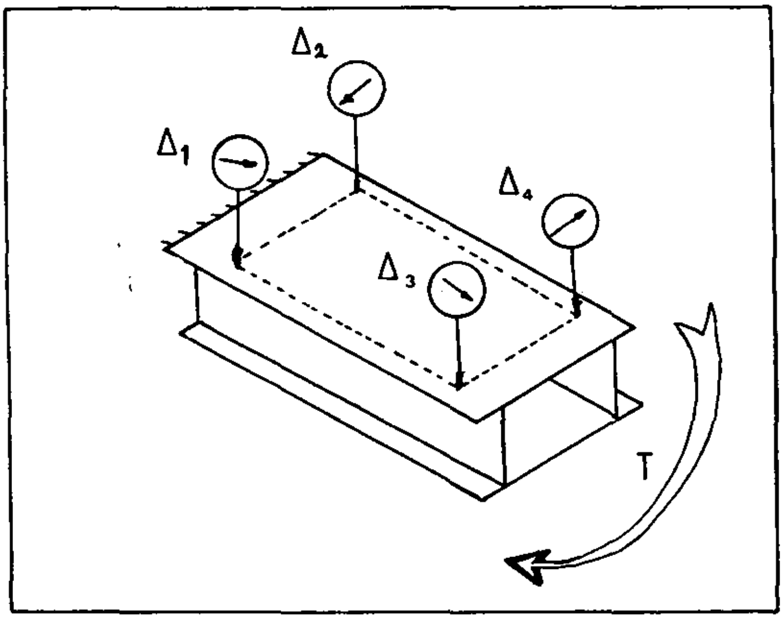


Figure 4.5 Shear flow path in adhesive.

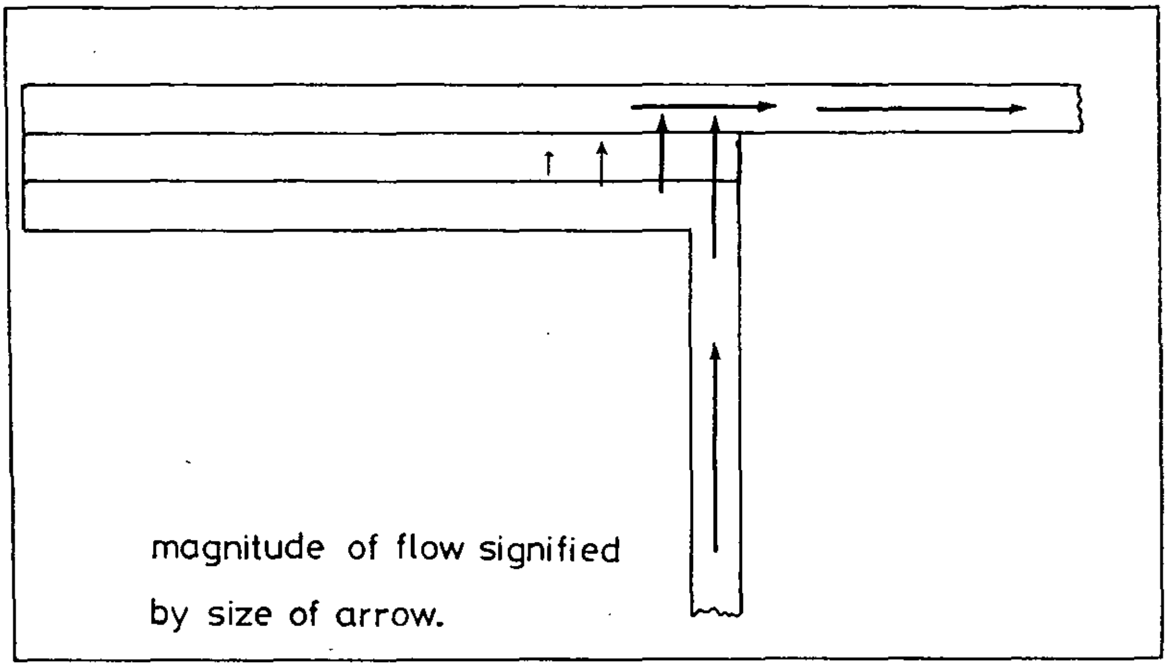


Figure 4.6 Plots of photo-stress results.

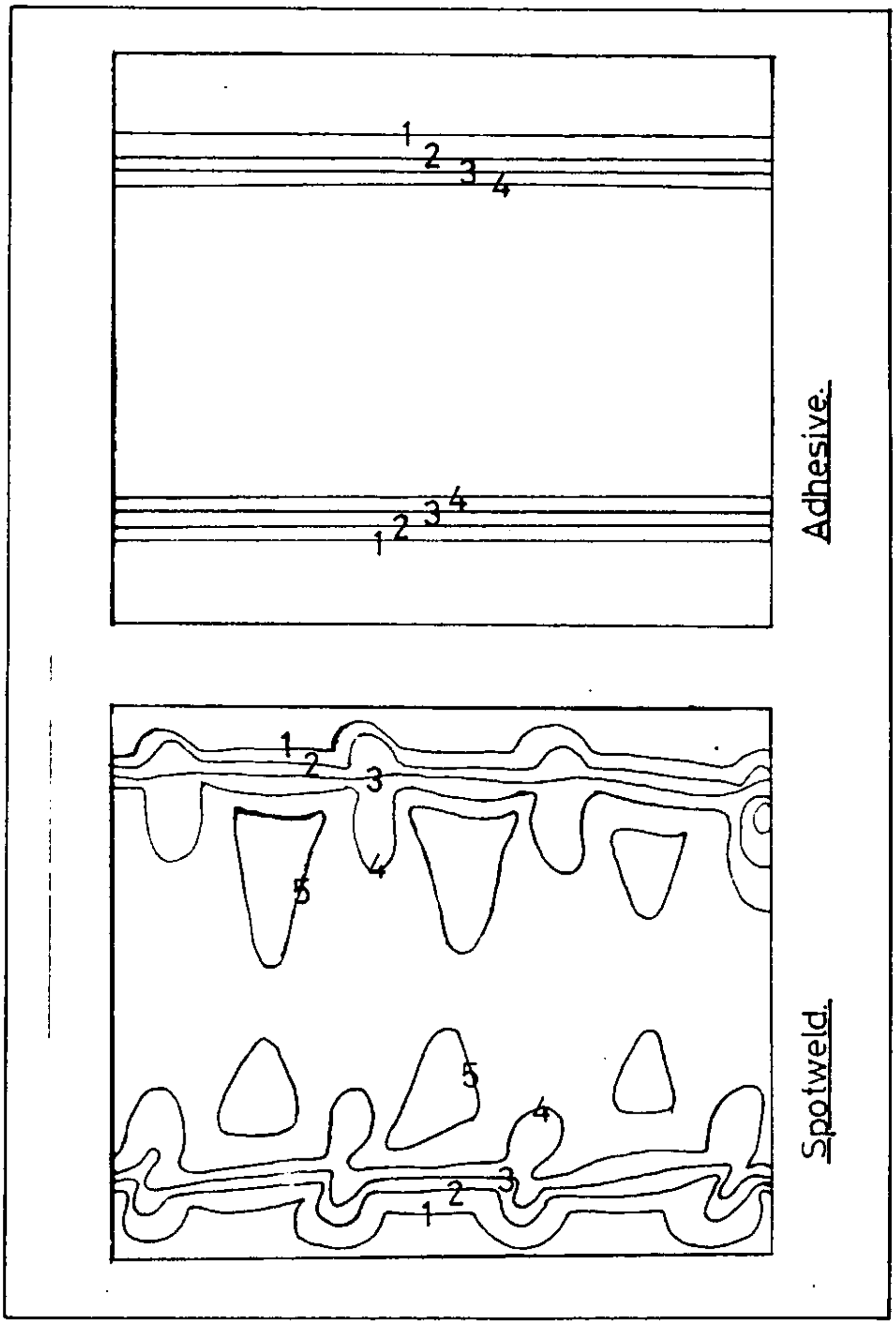
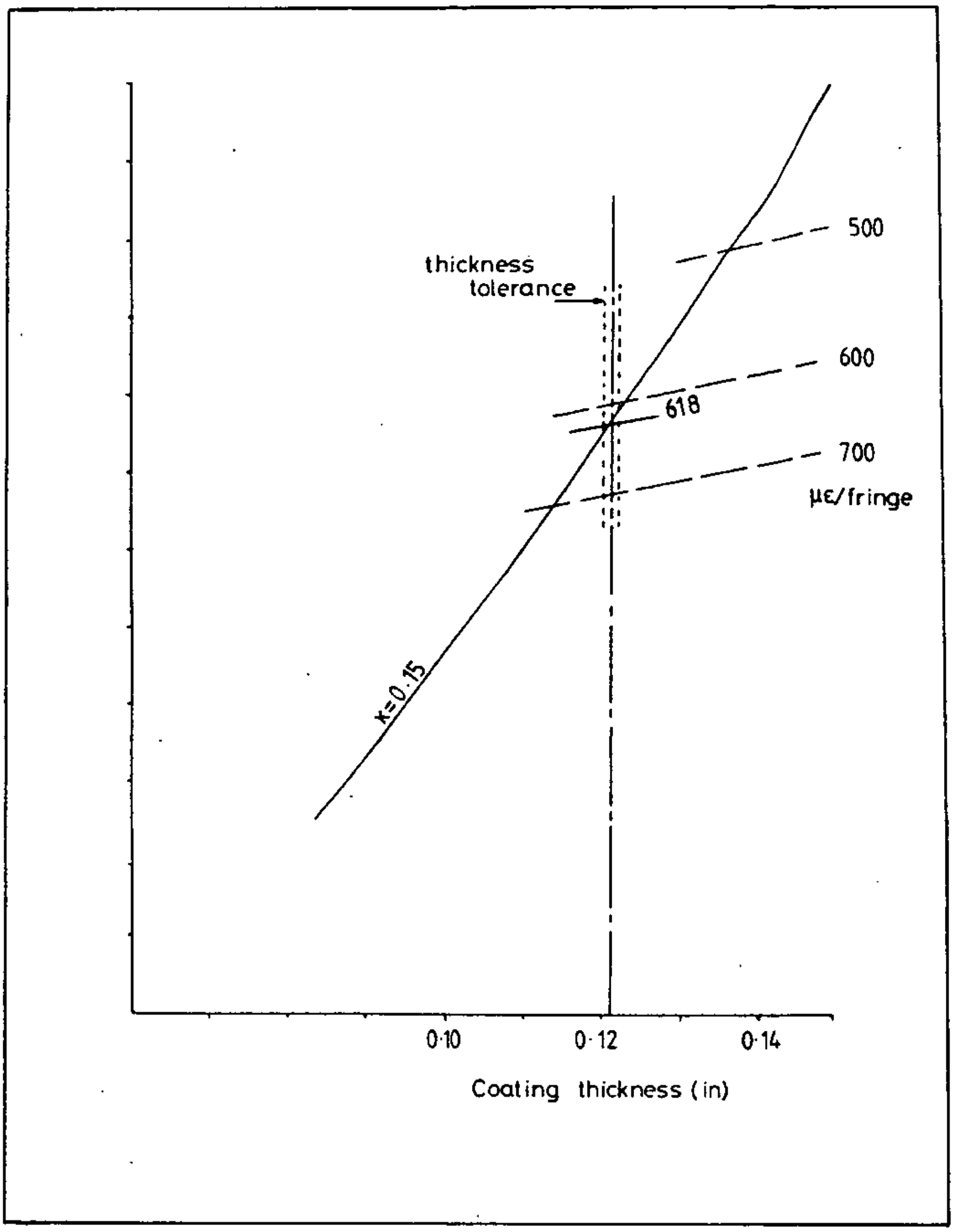


Figure 4.7 Calibration curve for birefringent coating.



Chapter 5.

Approximate analysis of adhesive joints using the shear lag equations.

Three different cases are to be discussed in this section:

1) The lap joint in tension.
2) The butt joint in shear.
3) A lap joint in shear due to beam bending.

5.1 The lap joint.

This is the simplest configuration to be discussed. It consists only of two overlapping plates with a layer of adhesive between them. The assumptions made in the analysis are as follows:

(i) The stress in the plates is pure tension and remains constant through the thickness of the plate.

(ii) The stress in the adhesive is pure shear and also remains constant through the adhesive thickness.

(iii) All bending effects are ignored.

(iv) All materials behave elastically under the applied load.

The geometry of the lap joint is shown in Figure 5.1.

Considering a small section taken from the joint (Figure 5.2). If plate 1 has moved by an amount U_1, and plate 2 by an amount U_2 we can define the strains in the plates as:

$$\boxed{\varepsilon_1 = \frac{dU_1}{dx} \quad , \qquad \varepsilon_2 = \frac{dU_2}{dx}}$$

applying Hooke's Law the tensions in the plates can be defined by:

$$\boxed{\begin{aligned} T_1 &= E_1 b t_1 \frac{dU_1}{dx} \\ T_2 &= E_2 b t_2 \frac{dU_2}{dx} \end{aligned} \qquad 1}$$

The adhesive layer will be subjected to a shear strain caused by the difference in the displacements of the plates on either side:

$$\gamma = \frac{u_2-u_1}{t_g} \quad , \qquad \tau = G\gamma = \frac{G}{t_g}(u_2-u_1) \tag{2}$$

For equilibrium of the elemental section through the joint at a distance x from one end (Figure 2) we obtain:

$$\begin{aligned} T_1 + \frac{dT_1}{dx}dx + \tau b dx - T_1 &= 0 \\ T_2 + \frac{dT_2}{dx}dx - \tau b dx - T_2 &= 0 \end{aligned} \tag{3}$$

giving :

$$\frac{dT_1}{dx} = -\tau b \qquad \text{and} \qquad \frac{dT_2}{dx} = \tau b \tag{4}$$

as a check;

$$\frac{dT_1}{dx} + \frac{dT_2}{dx} = 0 \text{ , integrating gives } T_1+T_2 = \text{constant} = T$$

From equation 2:

$$u_2-u_1 = \frac{\tau t_g}{G}$$

$$\Rightarrow \quad \frac{du_2}{dx} = \frac{du_1}{dx} = \frac{t_g}{G}\frac{d\tau}{dx} \quad \text{or} \quad \frac{d\tau}{dx} = \frac{G}{t_g}\left(\frac{T_2}{E_2bt_2} - \frac{T_1}{E_1bt_1}\right) \tag{5}$$

For all cases considered in this project $E_1=E_2=E$, $t_1=t_2=t$.

$$\frac{d^2\tau}{dx^2} = \frac{G}{t_gEbt}\left(\frac{dT_2}{dx} - \frac{dT_1}{dx}\right) = \frac{2G}{t_g.E.t}\tau$$

$$\frac{d^2\tau}{dx^2} - \lambda^2\tau = 0 \quad \text{where} \quad \lambda^2 = \frac{2G}{t_g.E.t} \tag{6}$$

The solution of this equation is of the form:

$$\tau = C_1\cosh\lambda x + C_2\sinh\lambda x \tag{7}$$

C_1 & C_2 must be found from boundary conditions:

$x=0$, $T_1=T$, $T_2=0$

$x=L$, $T_2=T$, $T_1=0$

To obtain the boundary conditions in terms of τ , not T, we use equation 5 which gives:

$$x = 0, \quad \frac{d\tau}{dx} = -\frac{GT}{E\ bt\ t_g}$$

$$x = L, \quad \frac{d\tau}{dx} = \frac{GT}{E\ bt\ t_g}$$

differentiating equation 7 gives

$$\frac{d\tau}{dx} = \lambda C_1 \sinh\lambda x + \lambda C_2 \cosh\lambda x$$

at $x=0$,

$$-\frac{GT}{E\ bt\ t_g} = \lambda C_2 \quad \therefore \quad C_2 = \frac{-GT}{E\ bt\ t_g \lambda}$$

at $x=L$,

$$\frac{GT}{E\ bt\ t_g} = \lambda C_1 \sin\lambda L - \frac{GT}{E\ bt\ g_g} \cosh\lambda L$$

$$\therefore \quad C_1 = \frac{GT}{Et\ t_g b\lambda \sin\lambda L} (1 + \cosh\lambda L)$$

which gives equation 6 as:

$$\tau = \frac{GT}{Et\ t_g b\lambda \sin\lambda L} [\cosh\lambda x + \cosh\lambda L \cosh\lambda x - \sinh\lambda L \sinh\lambda x]$$

$$\tau = \frac{GT}{Et\ t_g b\lambda \sin\lambda L} [\cosh\lambda x + \cosh\lambda (L-x)]$$

5.2 The butt joint in shear due to torsion.

This loading is produced in a joint, as shown in Figure 5.3, when the beam is subjected to a torque. The loading produced is a shear along the length of the beam and thus along the length of the joint seam.

This is a somewhat more complex situation than the planar lap joint since there is now a 3-dimensional stress field (although the stresses are invariant along the length of the seam).

The assumptions made in this analysis are as follows:

(i) The stress in the plates is pure shear and remains constant through the the thickness of the plate.

(ii) The stress in the adhesive is pure shear and remains constant through the thickness of the adhesive.

(iii) All bending effects are ignored.

(iv) All materials behave elastically.

The geometry of the joint is shown in Figure 5.4.

At any point in the adhesive we can say that the shear strain in the adhesive will be the difference in the warping movement of the two panels divided by the thickness of the adhesive:

$$\boxed{\frac{w_1 - w_2}{t_g} = \gamma = \frac{\tau_2}{G}} \qquad 1$$

since the joint is symmetrical (i.e. $W_1 = -W_2 = W$)

$$\boxed{\frac{2w}{t_g} = \frac{\tau_2}{G_g}} \qquad 2$$

The shear strain in the metal panels is given from the rate of change of warping with distance:

$$\boxed{\gamma = \frac{dw}{dz} \quad , \quad \tau_1 = G_m \frac{dw_1}{dz} = G_m \dot{w}} \qquad 3$$

Taking the stress distribution on an element to be as shown in Figure 5.5, the force equilibrium on the element will be:

$$\boxed{\tau_1 t_m - (\tau_1 + \frac{d\tau_1}{dz} dz) t_m = \tau_2 dz} \qquad 4$$

i.e. the force on the bottom of the element minus the force on the top of the element must equal the force applied to the adhesive.

$$\therefore \quad \frac{d\tau_1}{dz}\, dz\, t_m = \tau_2 dz$$

or
$$\frac{d\tau_1}{dz} = \tau_2/t_m \qquad 5$$

differentiating the warping equation 5.2.2 gives:

$$\frac{d\tau_2}{dz} \; \frac{t_g}{G_g} = 2\,\frac{dw}{dz}$$

and substituting into equation 5.2.4 gives:

$$\frac{d^2\tau_1}{dz^2} \; \frac{t_m t_g}{G_g} = 2\,\frac{dw}{dz}$$

combining with equation 5.2.3 gives a differential equation relating $\frac{d^2\tau}{dz^2}$ with τ :

$$\frac{d^2\tau_1}{dz^2} = \frac{2G_g}{t_m\, t_g\, G_m}\,\tau_1 \qquad 6$$

The solution of this equation is of the form:

$$\tau_1 = A\cosh kz + B\sinh kz \text{ , where } k = \sqrt{\frac{2G_g}{G_m\, t_m\, t_g}}$$

at
$$z = 0, \quad \tau_1 = \tau \quad \Rightarrow \quad A = \tau$$
$$z = W, \quad \tau_1 = 0 \quad \Rightarrow \quad \tau \cosh kw + B\sinh kw = 0$$
$$\Rightarrow \quad B = -\tau\,\frac{\cosh kw}{\sinh kw}$$

Thus:
$$\tau_1 = \tau\left[\cosh kz - \frac{\cosh kw}{\sinh kw}\,\sinh kz\right] \qquad 7$$

and from equation 5.2.4:

$$\tau_2 = \tau\, t_m\, k\left[\sinh kz - \frac{\cosh kw}{\sinh kw}\,\cosh kz\right] \qquad 8$$

and thus:

$$\boxed{w = \frac{\tau_2 t_g}{2G_g}}$$

Equation 8 gives the stress across the adhesive at any point across the width of the joint. Plots of this stress for different adhesive/metal combinations are given in Figure 5.6.

5.3 The effect of an adhesive joint on the top and bottom faces of a beam.

Figure 5.7 shows the assumed layout of the fabricated beam and Figure 5.8 shows a small section through the beam at a distance x from one end. All assumed forces and deflections are shown in the Figure.

The assumptions made in this analysis are:

(i) The top and bottom plates are subject to a tensile force only. which remains constant through their thickness.

(ii) The beam is in pure bending so the web is subject to a bending moment only.

(iii) The stress in the adhesive is pure shear and remains constant through its thickness.

(iv) All materials behave elastically.

(v) The load is applied to the web only.

(vi) There is no variation in stress over the surface of the closing plates.

(vii) The beam is built in at one end and totally free at the other end where the moment is applied.

The governing equations may be deduced from Figure 5.8, and are as follows:

$$\gamma = \frac{w_2 - w_1}{t_g} = \frac{\tau}{G} \qquad 1$$

$$\frac{dP_1}{dx} = -\tau a \qquad \left(\frac{d^2P_1}{dx^2} = -\frac{d\tau}{dx} a\right) \qquad 2$$

$$P_1 = AE\frac{dw_1}{dx} \tag{3}$$

also for equilibrium: (M_T= total moment applied)

$$M_T = 2P_1h + M_w \tag{4}$$

Differentiating equations 1 and 2 gives:

$$\frac{dw_2}{dx} - \frac{dw_1}{dx} = \frac{t_g}{G}\frac{d\tau}{dx} = -\frac{t_g}{G_a}\frac{d^2P_1}{dx}$$

Substituting into equation 3 to obtain:

$$\frac{dw_2}{dx} - \frac{P_1}{AE} = -\frac{t_g}{G_a}\frac{d^{2P}}{dx} \tag{5}$$

From elementary beam theory:

$$\frac{M}{I} = \frac{\sigma}{y}$$

so for the web alone:

$$\frac{M_w}{I_w} = \frac{\sigma}{y}$$

thus, on the top surface of the web:

$$\sigma = \epsilon E = \frac{M_w}{I_w}h$$

giving:

$$\frac{dw_2}{dx} = \frac{M_w}{EI_w}h \tag{6}$$

substituting equation 5.3.6 into equation 5.3.5 gives:

$$\frac{M_wh}{EI_w} - \frac{P_1}{AE} = -\frac{t_g}{G_a}\frac{d^2P_1}{dx^2}$$

from equation 5.3.4 we know that:

$$M_w = M_T - 2P_1h$$

thus:

$$\frac{(M_T-2P_1h)h}{EI_w} - \frac{P_1}{AE} = - \frac{t_g}{G_a}\frac{d^2P_1}{dx^2} = \frac{M_Th}{EI_w} - P_1\left[\frac{1}{AE} + \frac{2h^2}{EI_w}\right]$$

$$\frac{d^2P_1}{dx^2} = - \frac{G_aM_Th}{t_g\ EI_w} + P_1\frac{G_a}{t_g}\left[\frac{1}{AE} + \frac{2h^2}{EI_w}\right] \qquad 7$$

Equation 7 has a solution of the form:

$$P_1 = A\ \sinh kz + B\ \cosh kx + \frac{M_Th\ G_a\ E^2AI_w\ t_g}{t_g\ EI_w(2h^2AE+EI_w)G_a}$$

$$P_1 = A\ \sinh kx + B\ \cosh kx + \frac{M_ThA}{2h^2A+I_w}$$

where

$$k = \sqrt{\frac{G_a(2h^2AE+EI_w)}{t_g\ E^2\ AI_w}}$$

note, however, that:

$$I_T = 2h^2A + I_w$$

therefore :

$$P_1 = A\ \sinh kx + B\ \cosh kx + \frac{M_ThA}{I_T} \qquad 8$$

$$k = \sqrt{\frac{G_aI_T}{t_g\ EAI_w}}$$

5.3.1 Solving for A and B.

The boundary conditions are:

$$\text{at } x = 0,\ \tau = 0 \quad \frac{dP_1}{dx} = 0 \text{ (fixed end)}$$

$$0 = A\cosh 0 + B\sinh 0 \Rightarrow A = 0$$

$$\text{at} \quad x = L, \quad P_1 = 0$$

$$0 = B\cosh kL + \frac{M_T hA}{I_T} \quad \Rightarrow B = -\frac{M_T hA}{I_T \cosh kL}$$

$$\therefore \quad \underline{\underline{P_1 = \frac{M_T hA}{I_T}\left(1 - \frac{\cosh kx}{\cosh kL}\right)}}$$

As a check at this stage, we know that if the adhesive layer was solid, P_1 would be equal to:

$$\sigma A = \frac{M_T hA}{I_T}$$

from elementary beam theory.

At $x=0$, $k=\infty$. $$P_1 = \frac{M_T hA}{I_T}$$

and $$\frac{\cosh kx}{\cosh kL} \simeq 0$$ for large k, even when $x \approx L$. (Though not when $x=L$)

and for a very flexible adhesive where $k=0$, then $P_1=0$.

The beam deflections may be found by integrating the bending moment on the web.

from equation 5.3.4 we obtain:

$$M_w = M_T - 2P_1 h = M_T - \frac{2M_T h^2 A}{I_T}\left(1 - \frac{\cosh kx}{\cosh kL}\right)$$

$$M_w = M_T \left(1 - \frac{2h^2A}{I_T} + \frac{2h^2A \cosh kx}{I_T \cosh kL}\right)$$

(check; if k=0 then $M_w=M_T$)

Using standard beam theory.

$$\frac{d^2y}{dx^2} = \frac{M_w}{EI_w}$$

$$\frac{dy}{dx} = \frac{M_T}{EI_w}\left(x - \frac{2h^2A}{I_T}x + \frac{2h^2A \sinh kx}{k\, I_T \cosh kL}\right) + c$$

at $x = 0$, $\frac{dy}{dx} = 0$ therefore $c = 0$

$$y = \frac{M_T}{EI_w}\left(\frac{x^2}{2} - \frac{h^2A}{I_T}x^2 + \frac{2h^2A \cosh kx}{k^2 I_T \cosh kL}\right) + D$$

at $x = 0$, $y = 0$

$$0 = \frac{2h^2M_TA \cosh 0}{k^2 I_T EI_w \cosh kL} + D \quad \Rightarrow \quad D = \frac{-2h^2M_TA}{k^2 EI_T I_w \cosh kL}$$

Therefore the deflection;

$$y = \frac{M_T}{EI_w}\left(\left(1 - \frac{2h^2A}{I_T}\right)\frac{x^2}{2} + \frac{2h^2A}{I_T k^2 \cosh kL}(\cosh kx - 1)\right) \qquad 9$$

To obtain an efficiency rating fo the beam, we must compare the above deflection with the theoretical deflection from a homogeneous beam. For a homogeneous cantilever beam the deflection is:

$$(y_{max})_s = \frac{M_T \ell^2}{2EI}$$

Therefore it would be convenient for equation 9 to be of the form:

$$y = \frac{M_T}{2EI_T}(F(\ell^2))$$

now

$$y = \frac{M_T}{EI_T}\left(\left(\frac{I_T}{I_w} - \frac{2h^2A}{I_w}\right)\frac{x^2}{2} + \frac{2h^2A}{I_w k^2 \cosh kL}(\cosh kx-1)\right)$$

$$= \frac{M_T}{EI_T}\left(\frac{x^2}{2} + \frac{2h^2A}{I_w k^2 \cosh kL}(\cosh kx-1)\right)$$

$$y_{max} = \frac{M_T}{EI_T}\left(\frac{\ell^2}{2} + \frac{2h^2A}{I_w k^2}\left(1 - \frac{1}{\cosh kL}\right)\right)$$

efficiency,

$$\zeta = \frac{k}{k_s} = \frac{M_T/y_{max}}{M_T/(y_{max})_s} = \frac{(y_{max})_s}{y_{max}}$$

$$= \frac{\ell^2/2}{\frac{\ell^2}{2} + \frac{2h^2A}{I_w k^2}\left(1 - \frac{1}{\cosh kL}\right)}$$

$$\zeta = \frac{1}{1 + \frac{4h^2A}{I_w k^2 \ell^2}\left(1 - \frac{1}{\cosh kL}\right)}$$

This function is undefined when G=0, though it does have the expected limit value for G=0 of $\zeta = I_w/I$.

As an example let us consider the beam section as used in the experimental work in Chapter 4.

The nominal dimension are:

$h=0.02\text{m}$ $\qquad I_w=35\times10^{-9}\ \text{m}^4$

$l=0.45\text{m}$ $\qquad a=0.03\text{m}$

$G=10^9\ \text{N/m}^2$ $\qquad I_T=91\times10^{-9}\ \text{m}^4$

$E=71\times10^9\ \text{N/m}^2$ $\qquad tg=0.0005\text{m}$

$A=70\times10^{-6}\ \text{m}^2$

$$\therefore \quad k = \sqrt{\frac{1\times10^{9}\times.03\times91\times10^{-9}}{.0005\times71\times10^{9}\times70\times10^{-6}\times35\times10^{-9}}}$$

$$= \underline{177.2}$$

$$\zeta = \frac{1}{1+\dfrac{4\times.02^{2}\times70\times10^{-6}}{35\times10^{-9}\times177.2^{2}\times.45^{2}}\left(1-\dfrac{1}{\cosh(177.2\times.45)}\right)}$$

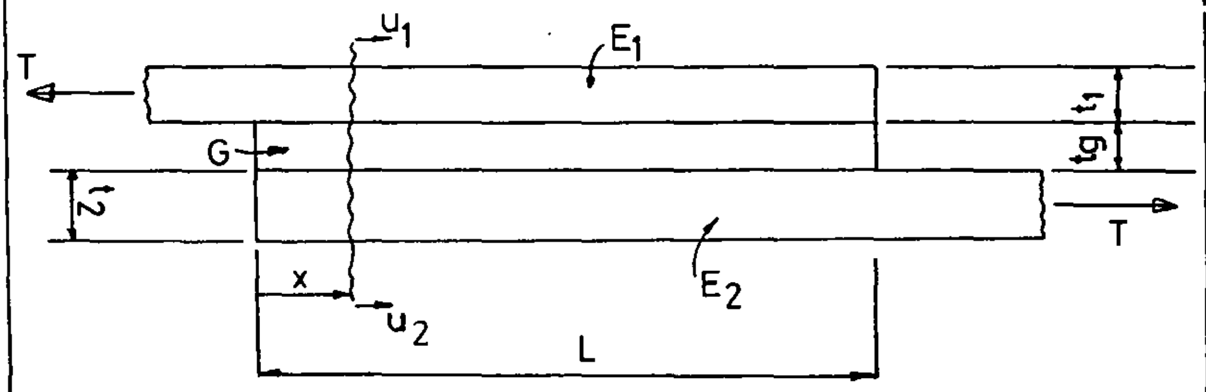
$$\zeta = \frac{1}{1+503\times10^{-6}\ 1} = \underline{\underline{1}}$$

Under these conditions the adhesive has no effect on the expected bending stiffness. If a is effectively only equal to the thickness of the metal (0.001m) and tg is increased to 0.001m, then:

$$k = 32.3$$

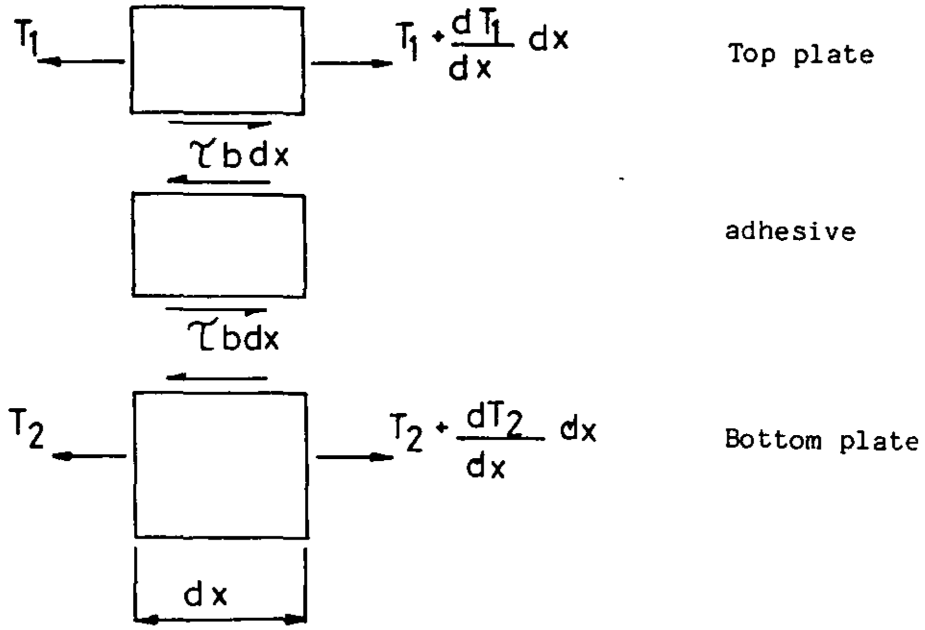
$$\zeta = 98.5\%$$

Figure 5.1 Lap joint geometry



Lap joint geometry and dimensions.

Figure 5.2 Section of lap joint



Stress conditions on an elemental section through a lap joint at a distance x from one end.

Figure 5.3 Shear loading in butt joint.

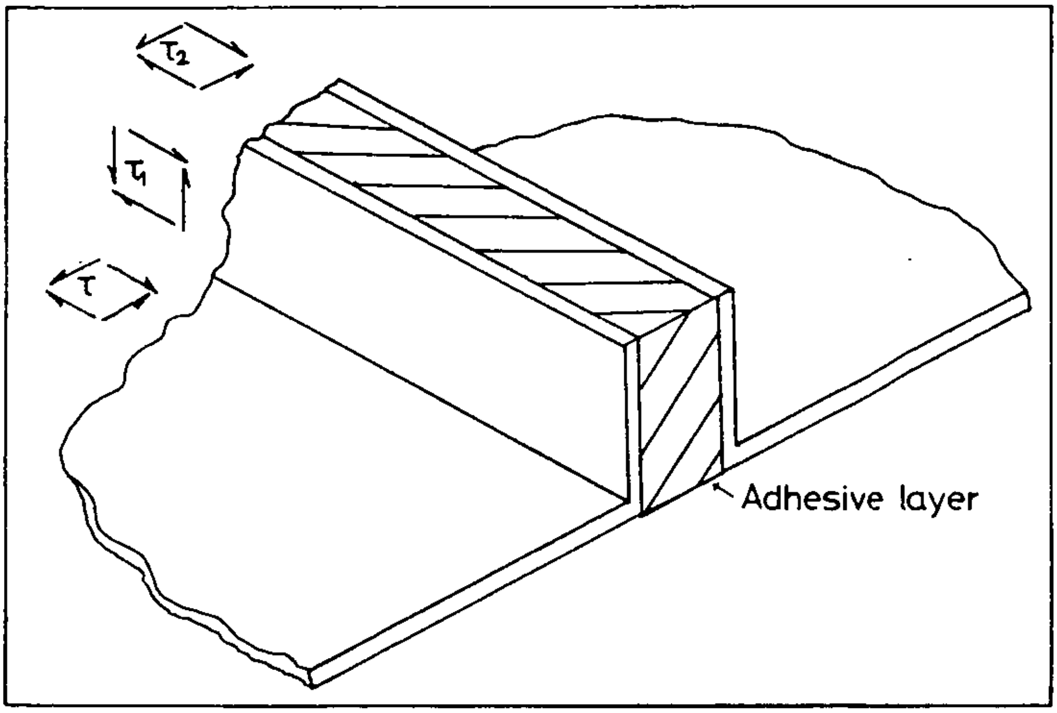


Figure 5.4 Butt joint geometry.

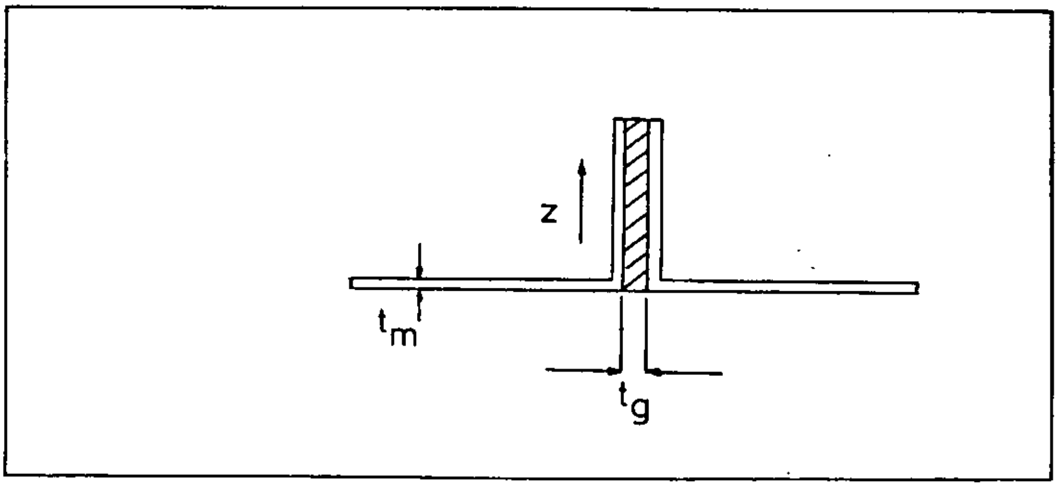


Figure 5.5 Stress distribution in butt joint.

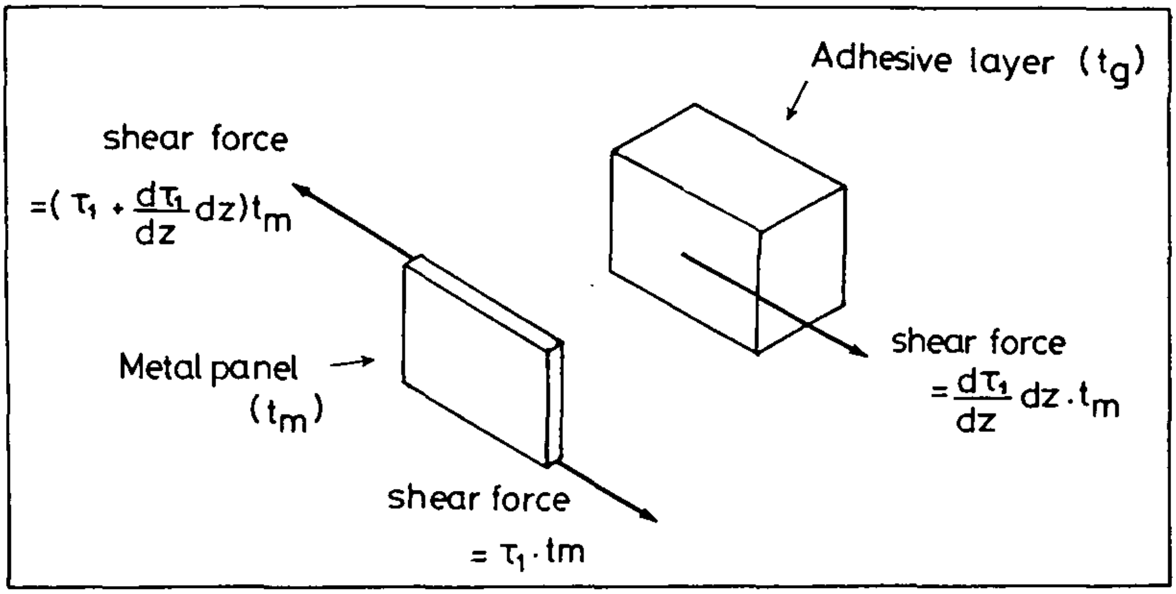


Figure 5.6 Plot of stress in joint.

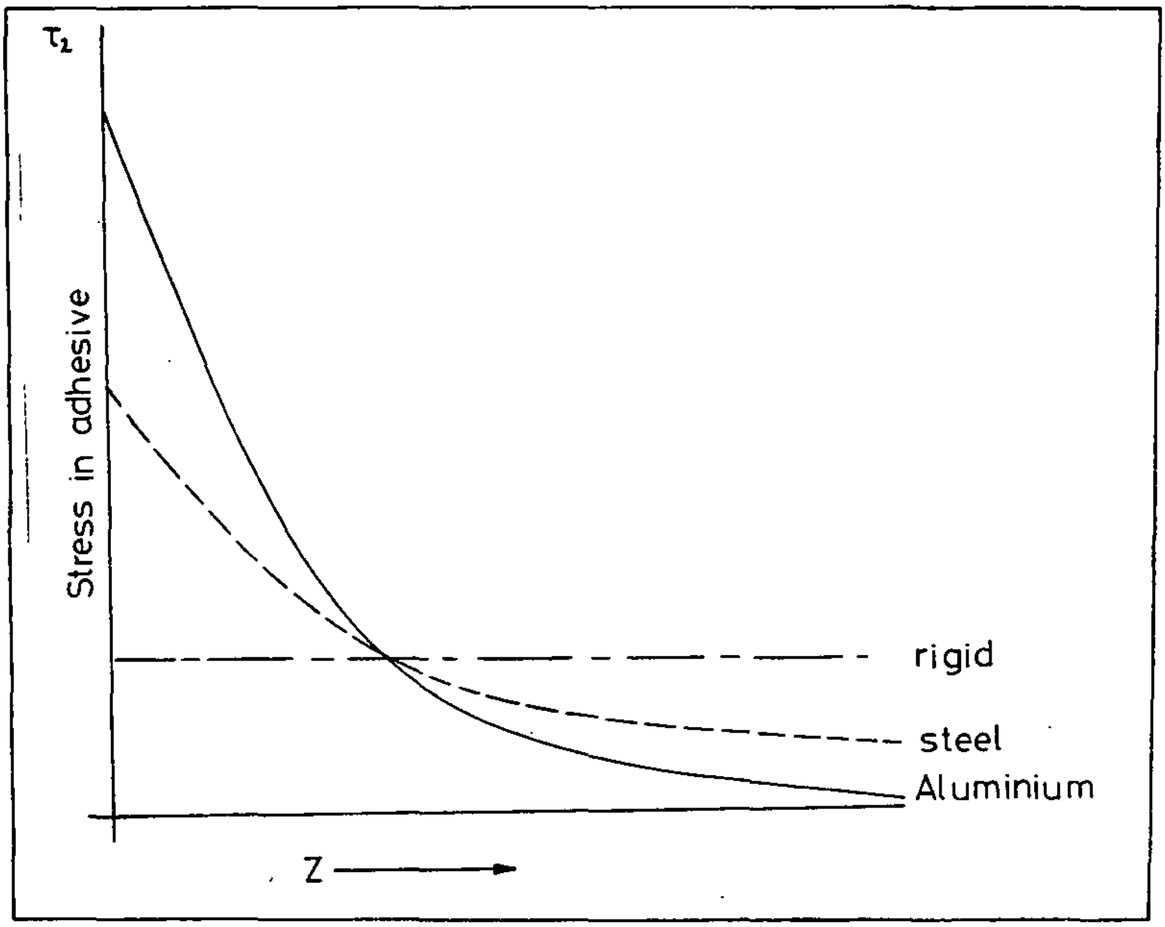


Figure 5.7 Beam geometry.

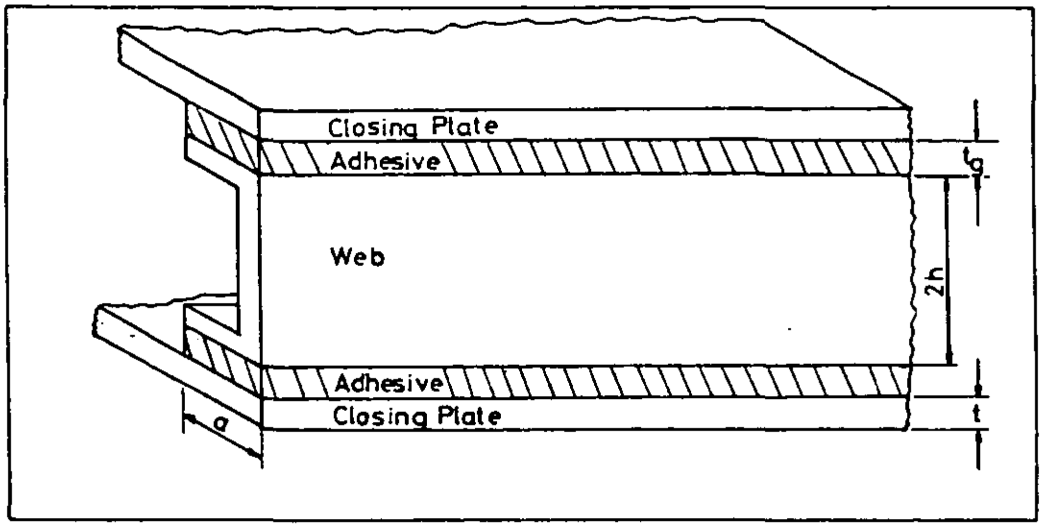


Figure 5.8 Section from beam.

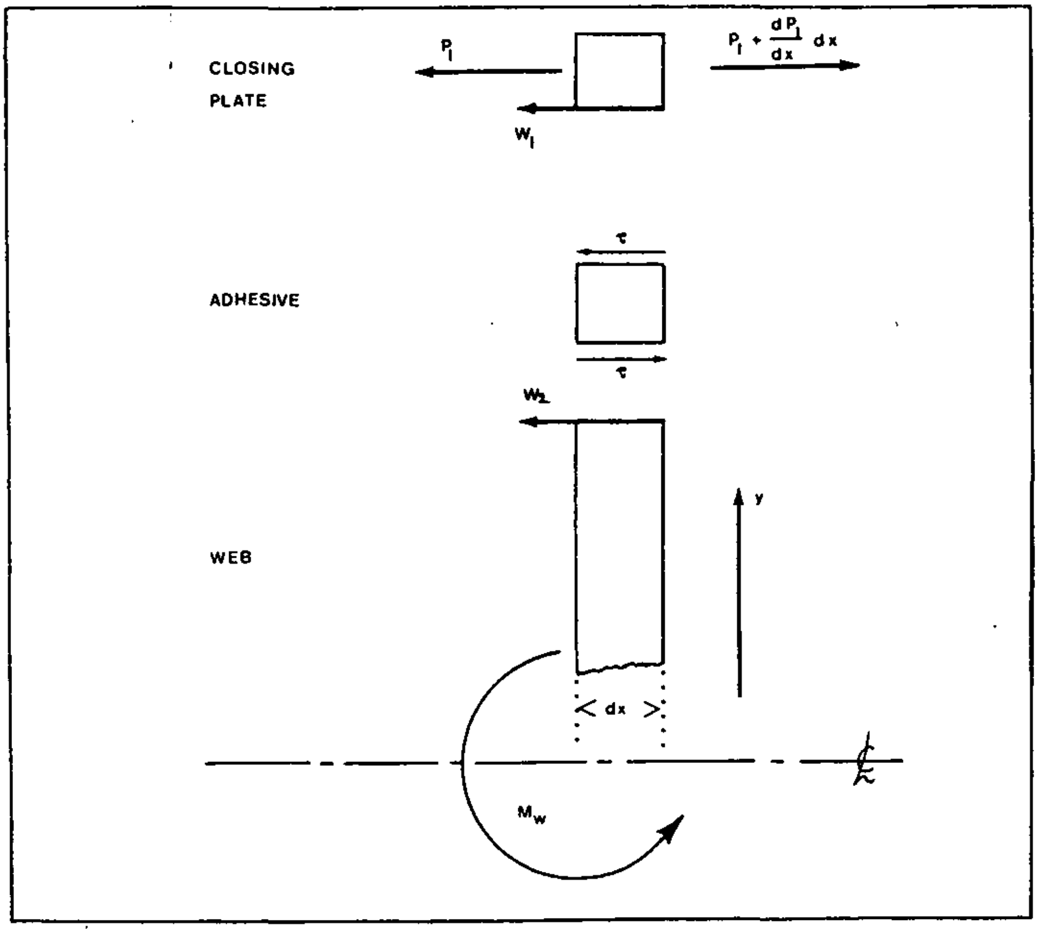
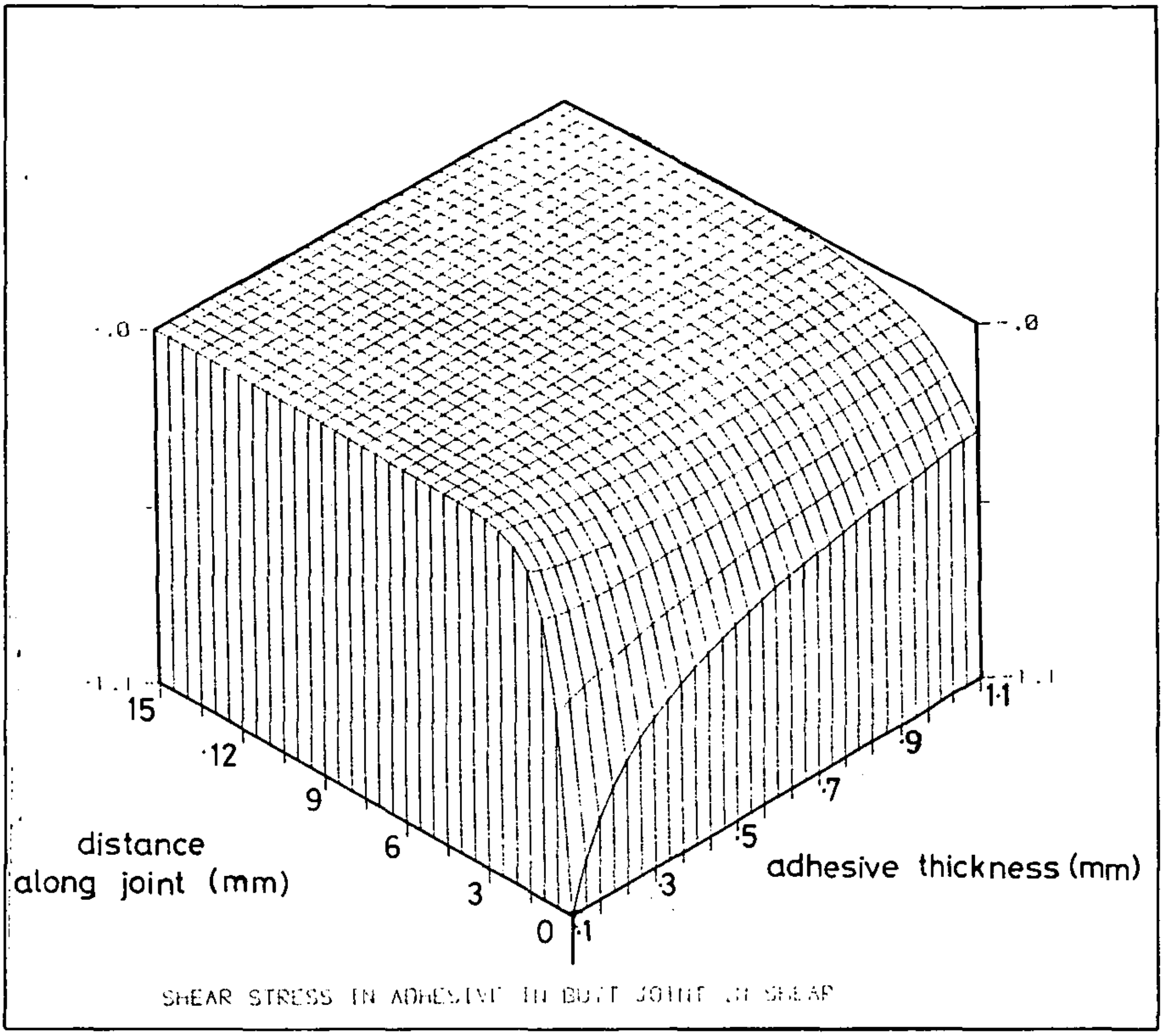


Figure 5.9 Carpet plot of stress.



Chapter 6.

Analysis of beams in Torsion.

It is commonplace to use the thin shell Batho-shear equations to calculate the torsion constant of closed section beams for use in finite element analyses of beam assemblies. These equations ignore the effects of warping constraints. This can usually be justified for closed section beams as they do not warp appreciably, though open sections (not covered by this analysis) are considerably influenced by warping constraints. The effect on the torsion constant of the fabrication technique cannot be precisely accounted for using the Batho-shear equations since assumptions made in the analysis do not apply.

6.1 Approximate analysis of closed sections.

It is assumed that there are no bending stresses or axial stresses present (implying that there is no warping constraint). Also since the thickness of the material is considered small, the shear stress has to be tangential to the surface everywhere, and does not vary through the thickness of the material. Figure 6.1 shows the shear flow $\{ \mathbf{q} = \tau t \}$ on two sections taken from a beam in torsion.

The shear flow on face AC produces complementary shear flows on AB and CD. Similarly, the shear flow CE produces complementary shear flows on CD and EF. Since the shear flow on CD is common, it must be equal. Therefore around any closed cylinder under pure torsion the shear flow is constant - the shear stress is therefore inversely proportional to the thickness.

The moment of any shear flow about any position (Figure 6.2) is:

$$\boxed{dT = qr\ dp}$$

since $\qquad r\,dp = 2 \times$ area enclosed

then $$T = 2\int q\,dA \qquad 1$$

q is constant, therefore:

$$T = 2Aq \qquad 2, \qquad \text{also } \tau = T/2At \qquad 3$$

To find J it is required to know the angle of twist, Θ, produced by the torque, T. Consider the energy absorbed by an element of dimensions dx,dy,dz. (Figure 6.3) :

$$dU^* = \frac{1}{2}\tau t\ dx\ \gamma dy$$

since $\gamma = \tau/G$

$$dU^* = \frac{\tau^2}{2G}\ t\ dxdy$$

In terms of an element of length L and dp wide, with a shear flow of q:

$$dU^* = \frac{q^2L\ dp}{2GT}$$

and for the whole cross-section:

$$U^* = \frac{q^2L}{2G}\oint\frac{dp}{t} \qquad 4$$

The work done by the torque T is $\frac{1}{2}T\theta$

$$\theta = \frac{q^{2}L}{GT}\oint\frac{dp}{t} \quad , \quad \text{since } T = 2Aq$$

$$\theta = \frac{qL}{2AG}\oint\frac{dp}{t} \qquad 5$$

Comparing these results with the elementary torsional equation T/J=GΘ/L, implies that the torsion constant is given by:

$$\underline{\underline{J = 4A^{2}/\oint\frac{dp}{t}}} \qquad 6$$

This equation is simple to apply and gives accurate results when the thickness is small in comparison to the size of the cross-section and any seam joints are assumed to be perfectly rigid. However, a problem occurs when the method of fabrication of the beam has to be taken into account. Spotwelds pose little problem since the path of the shear flow is known. The analysis of the cross-section need not consider those parts of the structure through which the shear flow does not pass (Figure 6.4). Actually the shear flow is much more concentrated at the spotwelds since they are discrete connections and the above method assumes a continuous longitudinal connection. The true J value will therefore be slightly lower than that calculated (see Chapter 4).

The problem is more complex with adhesive joints since the shear flow spreads out unevenly through the adhesive layer. The shear flow through the adhesive cannot be assumed to be constant since (as shown in Section 5.2) a butt joint in shear exhibits a high stress concentration at the root. The upper bound for the value of J can be calculated assuming a constant shear flow in the adhesive, and the lower bound assuming the shear flow to pass through the root with an effective adhesive thickness equal to that of the metal. The actual dimensions of the joint area and the properties of the adhesive will dictate towards which bound any particular solution will tend.

6.2 To obtain a satisfactory solution for a beam with adhesive seam joints it becomes necessary to use the exact equation developed by St.Venant (18).

The assumptions made are:

i) The rate of twist is constant along the length of the section.

ii) The warping of corresponding points is identical at all sections.

iii) Projections of the cross-section on the x-y plane rotate as rigid bodies (taking the axis of the the beam to be the z-axis).

Figure 6.5 shows a bar in torsion; p' is initially vertically above the point p. When the torque T is applied, o'p' rotates through an angle $\dot{\theta}$ z, relative to op, about the vertical z-axis. If the coordinates of p' are (x,y) or in polar coordinates (r, α), then the displacement of point p' may be written as:

$U = -rz\dot{\theta}\sin(\alpha)$

$\quad = -yz\dot{\theta}$ (in the x-direction)

$V = rz\dot{\theta}\cos(\alpha)$

$\quad = xz\dot{\theta}$ (in the y-direction)

$W = \dot{\theta}w(x,y)$ (in the z-direction)

assuming small deflections. w(x,y) is the unit warping function, which will be defined later.

The strains given by these deflections are:

$$\left.\begin{aligned}
\varepsilon_x &= \frac{\partial U}{\partial x} = 0 \\
\varepsilon_y &= \frac{\partial V}{\partial y} = 0 \\
\varepsilon_z &= \frac{\partial W}{\partial z} = 0 \\
\gamma_{xy} &= \frac{\partial U}{\partial y} + \frac{\partial V}{\partial x} = -\dot{\theta} z + \dot{\theta} z = 0 \\
\gamma_{yz} &= \frac{\partial V}{\partial z} + \frac{\partial W}{\partial y} = \dot{\theta} \left(\frac{\partial w}{\partial y} + x\right) \\
\gamma_{xz} &= \frac{\partial W}{\partial x} + \frac{\partial U}{\partial z} = \dot{\theta} \left(\frac{\partial w}{\partial x} - y\right)
\end{aligned}\right\} \qquad 7$$

Putting these equations in terms of stress:

$$\sigma_x = \sigma_y = \sigma_z = \tau_{xy} = 0$$

$$\left.\begin{aligned}
\tau_{yz} &= G\dot{\theta} \left(\frac{\partial w}{\partial y} + x\right) \\
\tau_{xz} &= G\dot{\theta} \left(\frac{\partial w}{\partial x} + y\right)
\end{aligned}\right\} \qquad 8$$

For equilibrium:

$$\left.\begin{aligned}
\frac{\partial \tau_{xz}}{\partial z} = \frac{\partial \tau_{yz}}{\partial z} &= 0 \\
\frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} &= 0
\end{aligned}\right\} \qquad 9$$

From equation 6.2.8 we obtain:

$$\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} = 0 \qquad 10$$

which may be written as $\nabla^2 w = 0$. τ_{xz} and τ_{yz} may be expressed in terms of a single function Φ (the stress function) such that:

$$\left.\begin{aligned} \tau_{xz} &= \frac{\partial\phi}{\partial y} \\ \tau_{yz} &= -\frac{\partial\phi}{\partial x} \end{aligned}\right\} \qquad 11$$

to satisfy equation 9. Substituting these values into equation 7 gives:

$$\tau_{yz} = -\frac{\partial\phi}{\partial x} = G\dot{\theta}\left(\frac{\partial w}{\partial y} + x\right) \qquad 12$$

$$\tau_{xz} = \frac{\partial\phi}{\partial y} = G\dot{\theta}\left(\frac{\partial w}{\partial x} - y\right) \qquad 13$$

differentiating (12) w.r.t. x and (13) w.r.t. y gives the equation:

$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = -2G\dot{\theta}$$

$$\text{or} \qquad \underline{\nabla^2\phi = -2G\dot{\theta}} \qquad 14$$

6.3 Obtaining the torsion constant from the stress function.

The total force on the section in the x-direction is:

$$F_x = \iint \tau_{xz}\,dxdy = \iint \frac{\partial\phi}{\partial y}\,dxdy$$

$$F_x = \int \left(\int_{b_1}^{b_2} \frac{\partial\phi}{\partial y}\,dy\right)dx$$

where b_1 and b_2 are on the boundaries where Φ is constant

$$F_x = \int [\phi]_{b_1}^{b_2} dx = 0 \qquad \text{since } \phi(b_1) = \phi(b_2)$$

Similarly Fy is also zero.

The anticlockwise moment on the section is:

$$T_0 = \iint (\tau_{yz}.x - \tau_{xz}.y) dxdy$$

$$= \iint \frac{\partial\phi}{\partial x}.x\ dxdy - \frac{\partial\phi}{\partial y}.y\ dxdy$$

but,

$$\iint \frac{\partial\phi}{\partial x}.x\ dxdy = \int\left[[x\phi]_{b_1}^{b_2} - \int \phi dx\right]dy$$

$$\left\{[x\phi]_{b_1}^{b_2} = 0, \text{ since } \phi \text{ is constant at } b_1 \text{ and } b_2\right\}$$

$$= \iint \phi\ dxdy$$

Also

$$\iint \frac{\partial\phi}{\partial y}.y\ dxdy = \iint \phi dxdy$$

thus

$$T_0 = 2\iint \phi dxdy$$

Now

$$T = JG\dot{\theta} = 2\iint \phi dxdy$$

giving

$$\underline{\underline{J = \frac{2}{G\dot{\theta}}\iint \phi dxdy}} \qquad 15$$

6.4 Calculation of the shear stresses from the stress function.

As defined in equation 11, the stress function yields the two shear stresses by partial differentiation:

$$\tau_{xz} = \frac{\partial\phi}{\partial y}$$

$$\tau_{yz} = -\frac{\partial\phi}{\partial x} \qquad 11$$

The maximum shear stress acting will be the gradient of the function Φ, normal to the contours of constant Φ. (Figure 6.6)

$$\tau = \frac{\partial\phi}{\partial n} \qquad \text{see Fig. T6}$$

$$\tau = \sqrt{\left(\frac{\partial\phi}{\partial y}\right)^2 + \left(\frac{\partial\phi}{\partial x}\right)^2} \qquad 16$$

6.5 Calculation of the unit warping function from Φ.

The two equations:

$$\nabla^2\phi = -2G\dot{\theta} \qquad \text{and} \qquad \nabla^2 w=0$$

define the relationship between Φ and ω. Defining:

$$\phi = w-k(x^2+y^2)$$

differentiation twice w.r.t x and y gives:

$$\nabla^2\phi = \nabla^2 w -2k-2k$$

since $\nabla^2 w=0$, $\qquad \nabla^2 w = -4k = -2G\dot{\theta}$

therefore $\qquad k = \frac{G\dot{\theta}}{2}$

Thus the values of w(x,y) can be found from:

$$\boxed{w = \phi + \frac{1}{2} G\dot{\theta}(x^2+y^2) \qquad 17}$$

The St.Venant torsion equations can only be solved analytically for very simple cross-sections. Results are given by Timoshenko (18) for square, triangular, elliptical and hollow elliptical bars. Other solutions may be found in similar text books and mathematical journals. The form of equation 14 is identical to that of many potential flow field problems such as heat transfer by conduction and magnetic flux problems in 2-dimensions (15). When the problem is not one of the simple cases quoted it becomes necessary to use numerical methods.

Many workers have used different numerical techniques with success. In the past the solution method adopted by many was that of finite differences (14,19) though more recently the finite element method has become more popular. The suite of finite element programs available at Loughborough (PAFEC75) has a solution routine for heat conduction problems and Appendix 1 describes its modification to solve the torsion equations. Listings of the modified routines are also given with details of how to run the program. Checking runs are shown where results are compared with those of previous workers.

6.5 Variations in torsion constant of a beam with changes in adhesive properties.

The Batho-Bredt shear equations for the beam cross section as shown in Figure 4.1 yield the following torsion constants:

$J=76.57\times10^{-9}m^4$ for Adhesive

or $J=56.60\times10^{-9}m^4$ for Spotwelds

The St.Venent results are given in table 6.1.

This study was undertaken to find the affect on the torsion constant caused by changes in the adhesive layer a) from changing the adhesive thickness and b) by changing the adhesive modulus.

A finite element model of the beam cross section was produced for use with the program described in section 6.4. The model was meshed as shown in Figure 6.7 with a considerably finer mesh on the flanges than that on the middle of the plates where the shear stress is known to be constant.

Initially the adhesive was given the same properties as the metal in the plates. This should give a result close to that obtained using the Batho-shear equations. The result obtained (table 6.1) is in fact 6% higher. This is accounted for by the effect of the flanges, which is ignored in the Batho-shear analysis.

During subsequent analyses the value of G for the adhesive was gradually reduced. Figure 6.8 shows the resulting fall off in J for the beam to be approximately linear. The gradient is also small with only a 4% change in beam stiffness for a more than 300% change in adhesive modulus. Equation 5.2.8 shows that the more flexible the adhesive layer, the more evenly distributed the shear stress becomes over the whole area. For the more rigid adhesives there is a sharp stress concentration at the root of the joint (see Figures 6.9,6.10). For this reason a flexible adhesive should be less prone to fatigue, without an associated drastic reduction in stiffness.

Glue line thickness variations are always likely to occur in production. What effect are these likely to have on stiffness? Here, the results show a more complex picture. Simply increasing the thickness of the adhesive increases the size of the beam making it stiffer. However, with the thicker adhesive layer it can be seen that there is a much more rapid drop off in beam stiffness as the adhesive modulus is reduced. Therefore to keep the stiffness constant with glue line thickness variations an adhesive/metal G ratio of about 14 is required. The use of flexible adhesives is not possible if wide variations in glue thickness are likely, due to the rapid fall in stiffness. Some compromise is required to keep the stress concentrations low while not allowing large stiffness variations to occur.

Table 6.1.

Values of J predicted by the St.Venent Torsion program

For beams similar to Figure 4.1.

Description Adhesive Thickness	 G ratio	Torsion constant J
.5mm	1	81.25
.5mm	15	79.40
.5mm	25	78.32
.5mm	50	75.32
.5mm	100	69.08
1.0mm	1	83.20
1.0mm	15	78.00
1.0mm	26	76.20
1.0mm	100	65.80

Figure 6.1 Complimentary shear flows on a beam in torsion

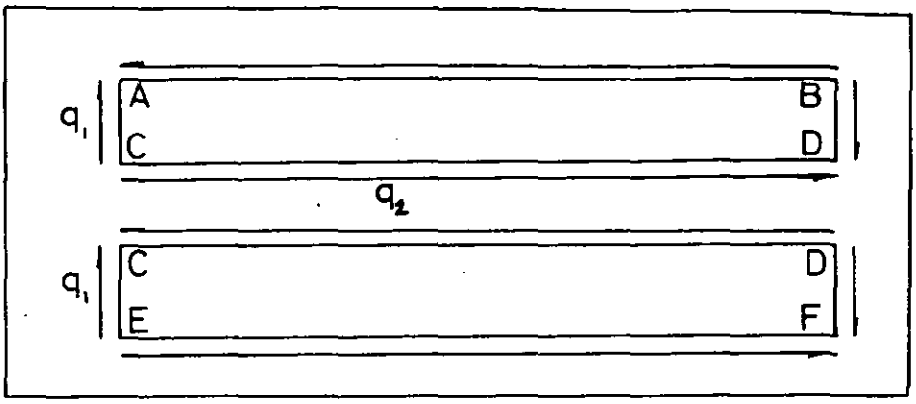


Figure 6.2 Moment due to shear flow.

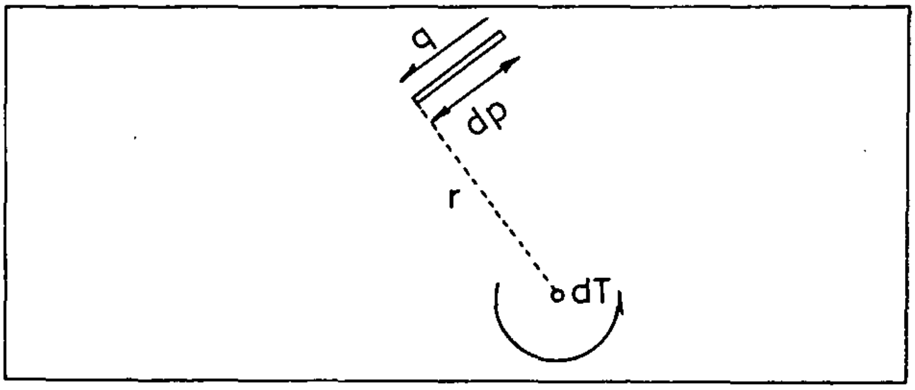


Figure 6.3 Energy absorption due to shear flow.

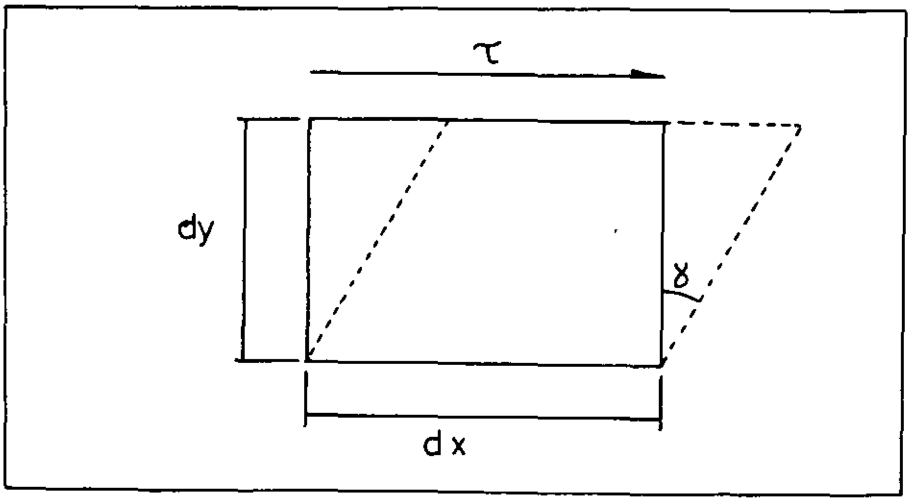


Figure 6.4 Shear flow path in spotwelded beam.

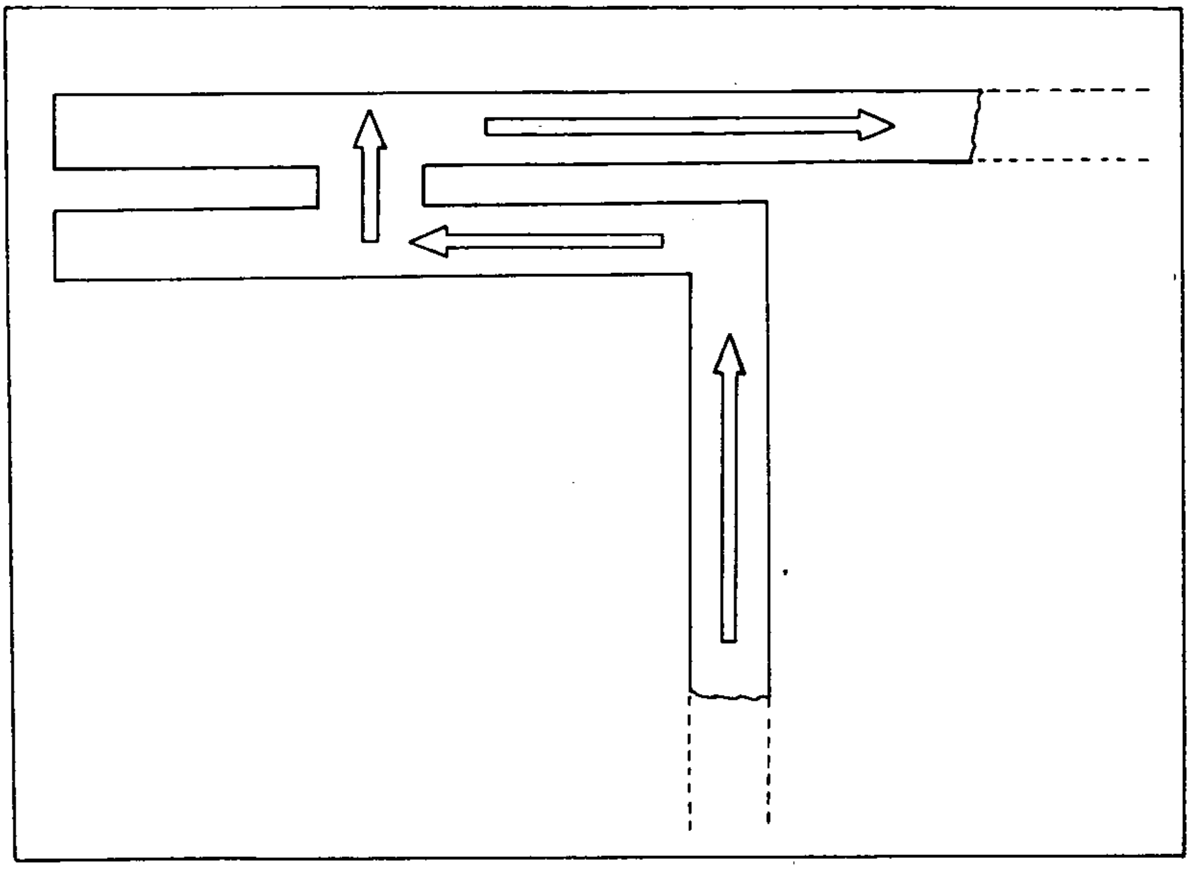


Figure 6.5 St.Venent torsion assumptions.

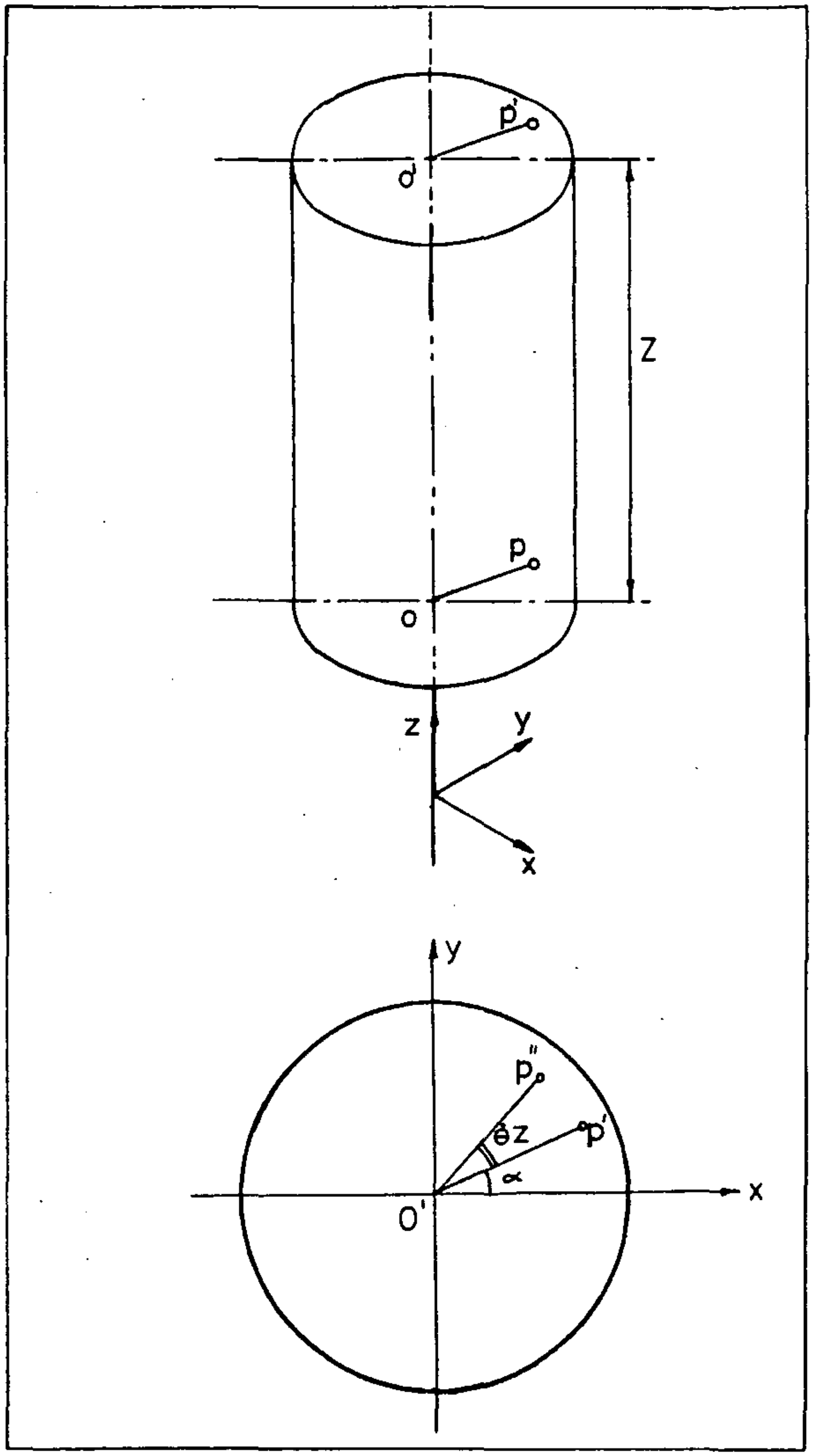


Figure 6.6 Absolute shear stress.

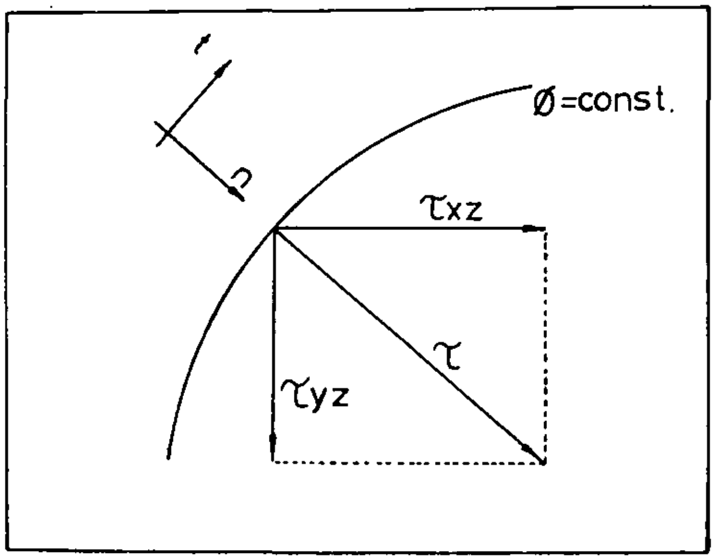


Figure 6.7 Finite element mesh of the beam cross section.

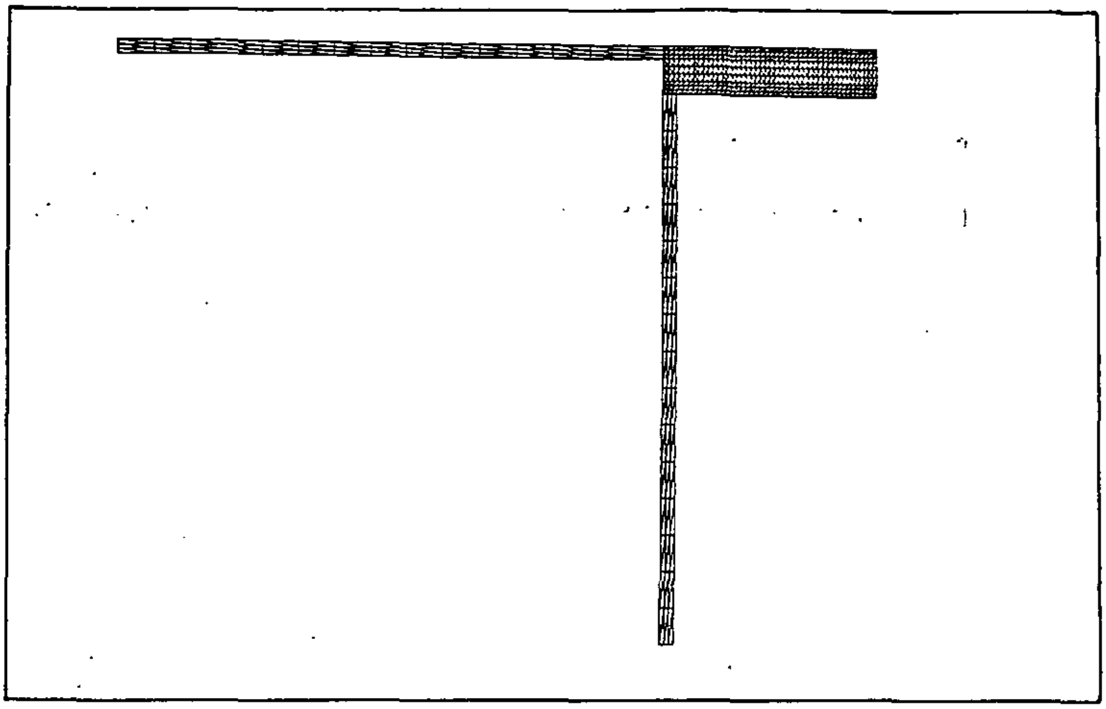


Figure 6.8 Graph of torsion constant v G ratio.

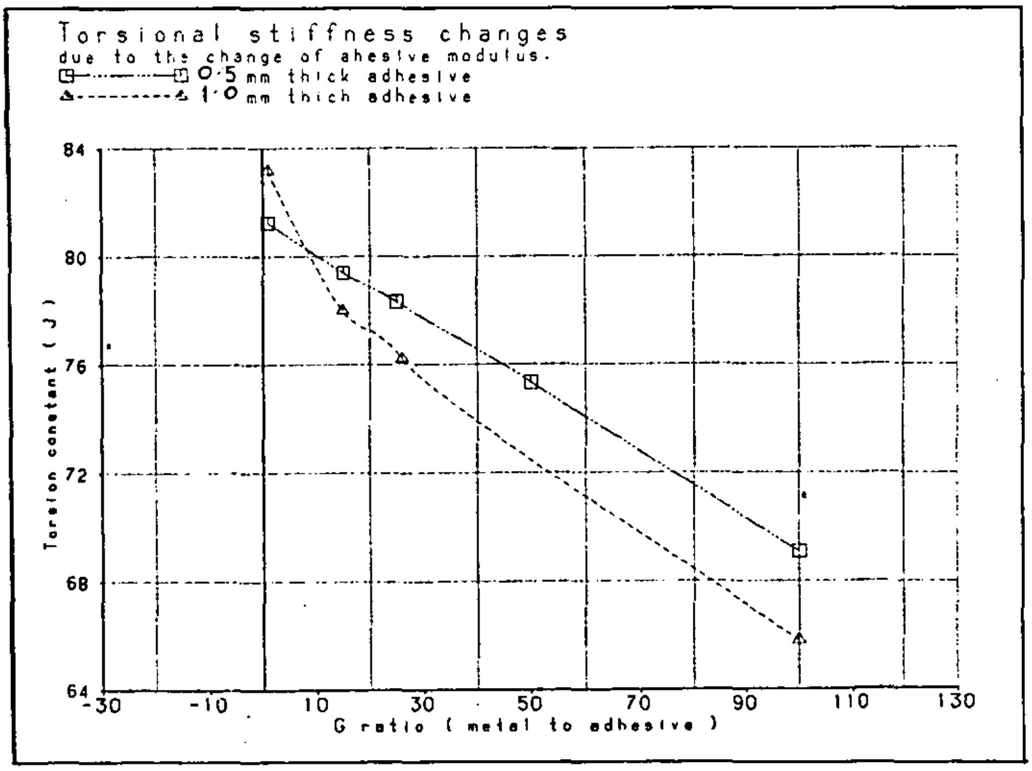


Figure 6.9 PAFEC75 plot of shear stress on section.

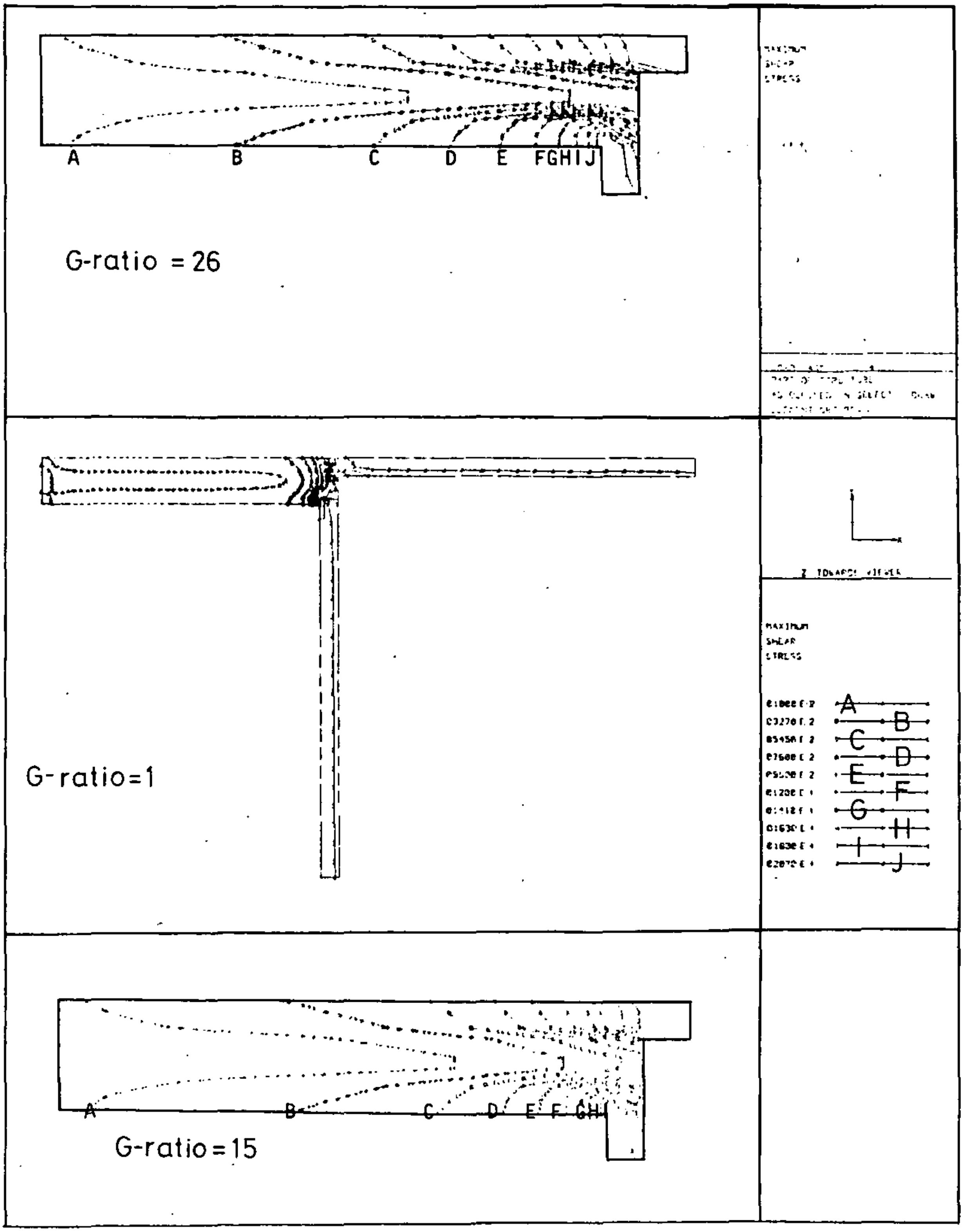
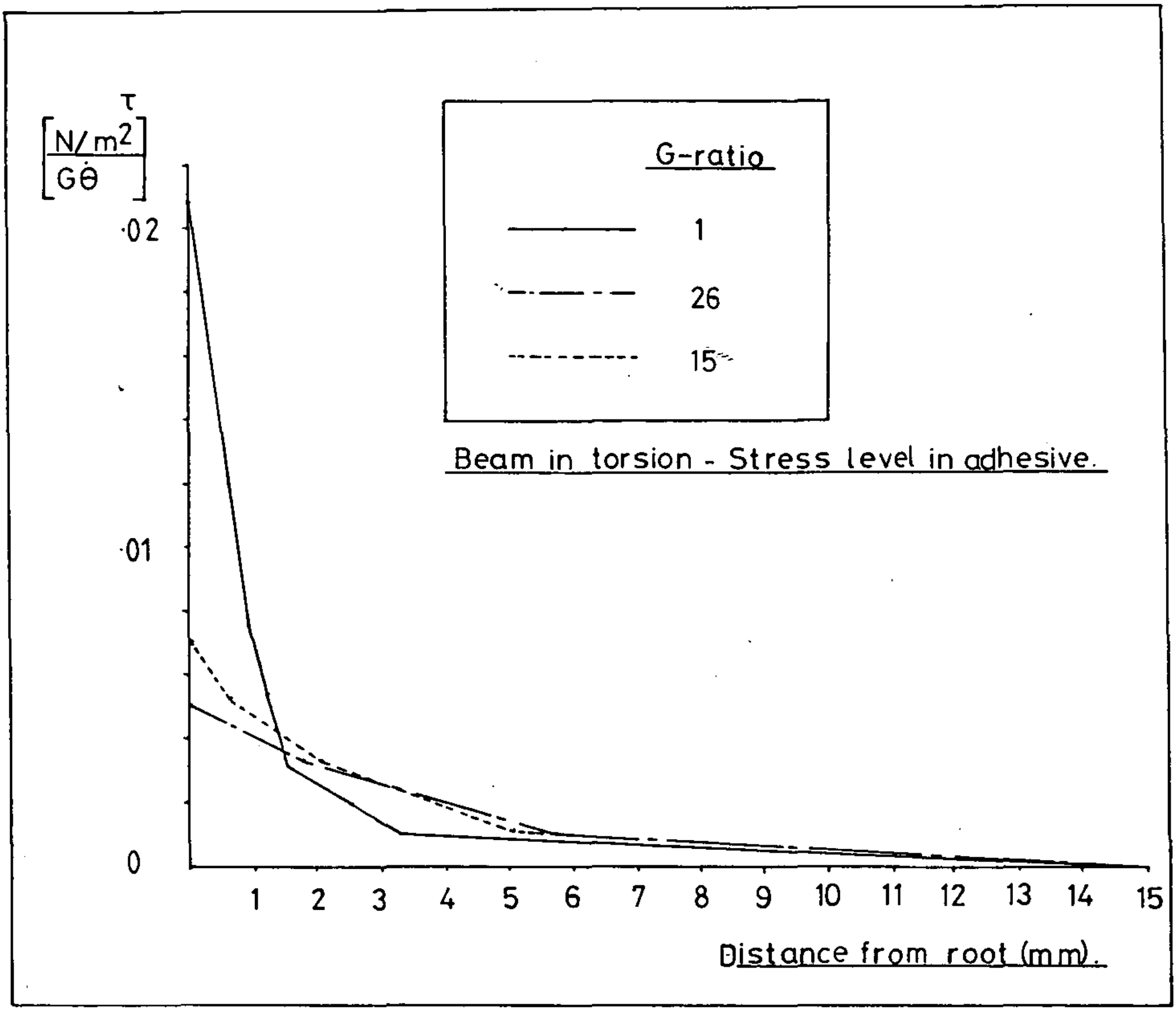
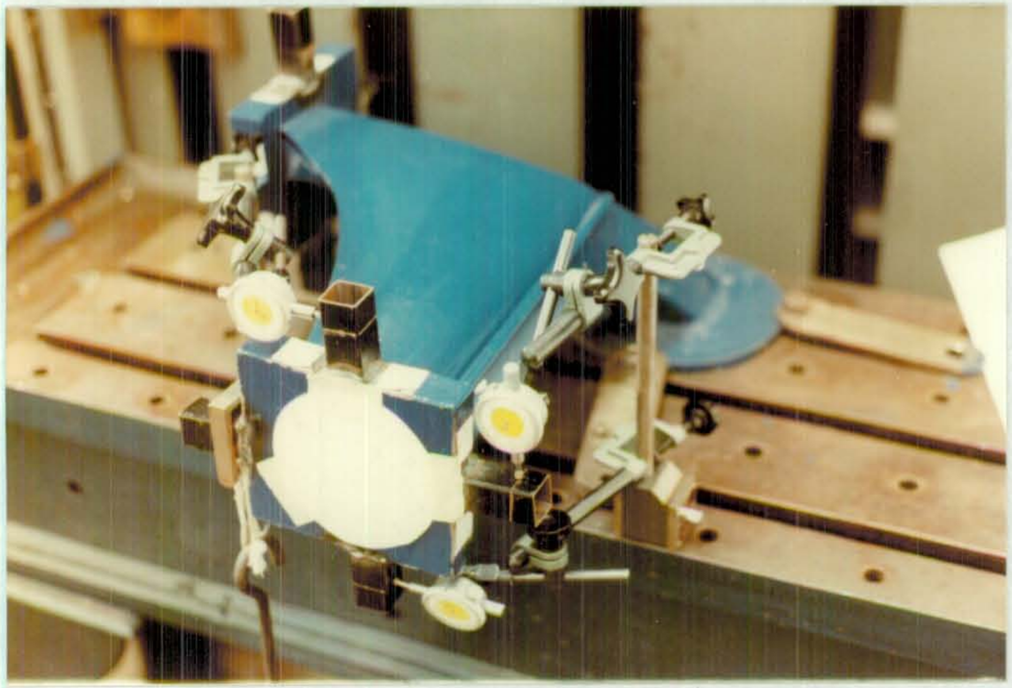


Figure 6.10 plot of adhesive stress against distance



Joint Stiffness Measurement


(a) End Diaphragm and Displacement Measurement

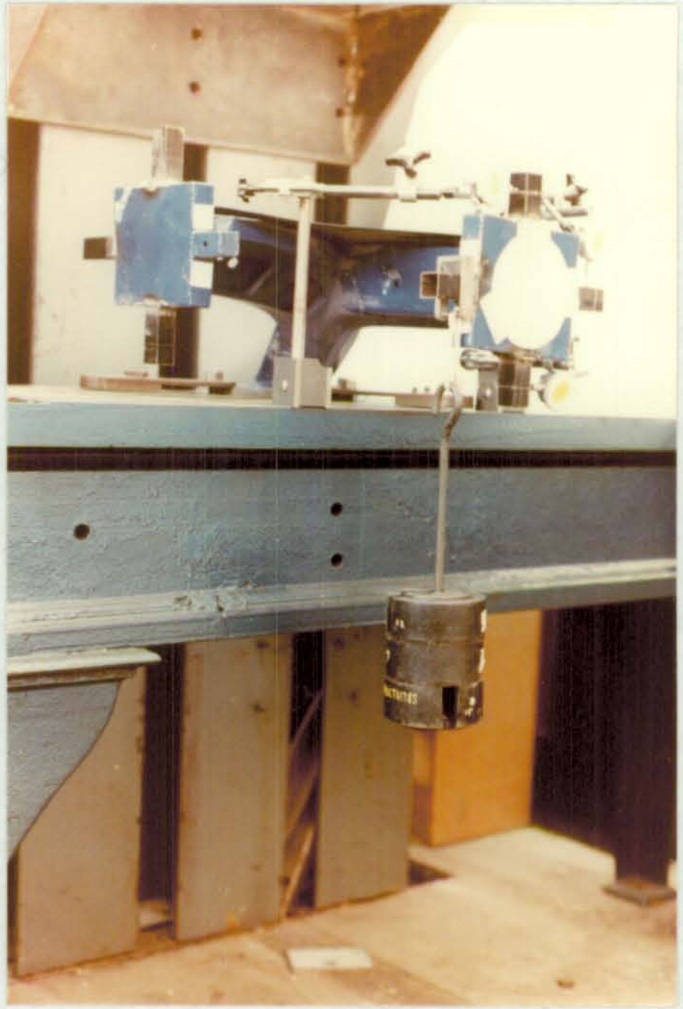

(b) Loading and Overall Layout

PLATE 3

Chapter 7.

Joint Flexibility and Stiffness.

It is well known (1,2) that the flexibility of the joints in a vehicle structure have a considerable effect on the overall stiffness of a vehicle. To account for this it is now common practice to develop complex finite element shell models of the major joints in a vehicle frame (Figure 7.1) to substitute as substructures into the full vehicle model. When used, these joint substructures reduce the overall vehicle stiffness by as much as 50% compared to the case for rigid joints. This considerably improves the correlation between the finite element results and those found from experimental measurements on vehicle bodies. However, the accuracy of the joint matrices should be compared with experimental results from the joint

The joints are very complex, being fabricated from several small pressings and joined by many spotwelds. The finite element joint models may use between 100 and 500 elements each, but is this enough for such a complex situation? To check their validity it was decided to measure the stiffness matrix of a joint cut from a vehicle. A similar piece of work had been undertaken by Sharman (3) on a simplified joint, which pointed to some of the problems likely to be encountered. Sharman used a T-joint (Figure 7.2) fixed at each end of the main beam but allowed to rotate around the y-axis. Forces were applied at the end of the T-beam.

The object was to obtain values for conceptual spring stiffnesses at the joint located between the two beams (Figure 7.3). Problems were encountered with non linear displacements due to the use of an indeterminate restraining system. The normal assumptions for spring joints also meant that it was implicit that there could not be any cross-coupling terms. To overcome these problems it was decided that in this experiment a determinate set of restraints should be used. Also, after initial studies of the possibility of using an arrangement of springs at the centre of the joint to define the flexibility, it was decided to simply produce a set of flexibility values for the outer ends of the joint beams. It is, however, a problem on a car body to decide precisely where the beam ends and the joint begins. The supposedly separate items merge gradually into each other so that an arbitrary decision has to be made to judge at which point along the beam the local joint distortions have diffused out into a set of deflections compatable with a beam element.

A three beam joint has 18 degrees of freedom and would therefore require 18x18=324 measurements to be made to be defined fully. This is obviously not feasible for an experiment to be performed manually. It is possible to reduce the quantity of readings required by assuming symmetry of the flexibility matrix, but 171 readings is still a large number of experimental measurements.

7.1 Forming a full stiffness matrix from a flexibility matrix.

Lively (ref. 40) gives a method using an equilibrium matrix to transform a reduced flexibility matrix into a full stiffness matrix. Starting with the basic stiffness definition in matrix form:

$$[R]=[K][U]$$

where [R] is the set of force vectors

and [U] is the set of displacement vectors at the ends of the joint beams. This may be partitioned to give:

$$\begin{bmatrix} R_1 \\ R_2 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad 2$$

R_1 is the set of forces at the restraints,

u_1 is the set of deflections at the restraints.

R_2 and u_2 are the unrestrained forces and deflections

For equilibrium:

$$H \times R_2 - R_1 = 0 \qquad 3$$

where [H] may be called an equilibrium matrix

From equation 7.1.2:

$$\begin{aligned} R_1 &= k_{11} u_1 + k_{12} u_2 \\ R_2 &= k_{21} u_1 + k_{22} u_2 \end{aligned} \qquad 4$$

During measurement $[U_1]$ is fixed and therefore zero. The deflections $[U_2]$ are non zero, and replacing these by the symbol d_2 thus: $[U_2]=[d_2]$ therefore:

$$R_2 = k_{22} d_2 \qquad 5$$

$[K_{22}]$ is the inverse of the measured flexibility $[F_{22}]$

From equation 7.1.3:

$$R_1 = -HR_2 = -H k_{22} d_2 \qquad 6$$

Now allowing for rigid body rotations, written as d^*;

$$\left.\begin{aligned} [U_1] &= [d_1^*] \\ [U_2] &= [d_2^*] + [d_2] \end{aligned}\right\} \qquad 7$$

where $[d_2^*] = [H]\ [d_1^*]$ (see Livesly) 8

Substituting for the rigid body motions from equation 7.1.7:

$$d_2 = u_2 - H^* u_1 \qquad 9$$

Substituting this into equations 7.1.5 and 7.1.6:

$$\left.\begin{aligned} R_1 &= -H\ k_{22}[u_2 - H^t u_1] = [H\ k_{22}\ H^t]u_1 - [Hk_{22}]u_2 \\ R_2 &= k_{22}[u_2 - H^t u_1] = -[k_{22}\ H^t]u_1 + k_{22} u_2 \end{aligned}\right\} \qquad 10$$

Comparing these with equation 7.1.2 gives:

$$\left.\begin{aligned}
[k_{11}] &\equiv [H\ k_{22}\ H^t] \\
[k_{12}] &\equiv -[H\ k_{22}] \\
[k_{21}] &\equiv -[k_{22}\ H^t] \\
[k_{22}] &= [k_{22}] = [F_{22}]^{-1}
\end{aligned}\right\} \qquad 11$$

7.2 Calculation of the equilibrium matrix [H].

The joint layout is shown in Figure 7.4, it has 3 nodes with node 1 fixed. For equilibrium all the forces must equal zero:

$$\begin{aligned}
X_1 + X_2 + X_3 &= \phi \\
Y_1 + Y_2 + Y_3 &= \phi \\
Z_1 + Z_2 + Z_3 &= \phi
\end{aligned} \qquad 12$$

also the moments about the fixed node must be zero:

$$\begin{aligned}
&(\text{z-axis}) \quad M_{z1}+M_{z2}+M_{z3}-X_2(x_2-x_1)-X_3(x_3-x_1)+Y_2(y_2-y_1)+Y_3(y_3-y_1) = \phi \\
&(\text{y-axis}) \quad M_{y1}+M_{y2}+M_{y3}-X_2(z_2-z_1)+X_3(z_3-z_1)-Z_2(x_2-x_1)-Z_3(x_3-x_1) = \phi \\
&(\text{x-axis}) \quad M_{x1}+M_{x2}+M_{x3}-Y_2(z_2-z_1)-Y_3(z_3-z_1)+Z_2(y_2-y_1)+Z_3(y_3-y_1) = \phi
\end{aligned}$$

If these relationships are taken as representing equation 7.1.3 then subtracting the $[R_1]$ terms leaves $[H][R_2]$, which can be written as shown in Figure 7.5. This, then, allows the measurement of the stiffness at the twelve degrees of freedom on the free arms of a 3-beam joint, enabling the full 18x18 square matrix for the whole joint to be produced. The [U] matrix is a square symmetric matrix [12x12] and requires measurements to be made of the displacement at each degree of freedom for each column of the force matrix. The force matrix is made up from 12 force vectors with all terms being zero exept for a single unit force on one degree of freedom. If symmetry of the displacement matrix is assumed it is then necessary to take a total of 12x(12+1)/2=78 measurements with 12 different load cases.

7.3 Experimental Apparatus.

The joint chosen for the experiment was the D-post / Cantrail / Rear-header rail joint from an Austin Metro, for which a finite element model was already available. This joint could be representative of any joint in a vehicle as it has no symmetry and does not lie in a plane. One of the arms of the joint has to be fully restrained. The one chosen was the rear header beam since it appeared to be the stiffest of the beams (being made from 3 panels) and is least likely to be effected by a torsion warping restraint as it is a closed section. It was therefore possible to simply braze this beam to an effectively rigid 1/2"-thick plate which could then be bolted securely to a test bench.

Some means also had to be devised for applying the forces and measuring the displacements on the ends of the other two beams. It was thought necessary to try to avoid any longitudinal restraint on torsion warping. To this end some box sections were devised which would be stiff in all directions but allow local deflections to occur along the axis of the beam by the relatively flexible action of bending in the plates (Figure 7.6). The cross pieces were used for the actual force application and displacement measurements. Initially it had been intended to apply pairs of opposing forces to obtain pure torsion, but a simpler solution appeared to be the application of two anti-symmetric forces on the ends of the cross pieces.When added these deflections equal those that would be producd by a single force of 2P at the centroid and when subtracted they represented torque of P.d about the centroid. (P is the applied force and d is the distance between the two forces).

The displacements were measured using dial gauges having a resolution of .001mm. Only six gauges were available, necessitating all loads to be applied a second time, with the DTI's moved to the remaining 6 positions. The forces were applied in increments of 20N to a maximum of 100N, and the displacements were measured at each increment as a check on linearity.

7.4 Acquisition of results.

The amount of calculation involved in working out the unit force displacements for each of the 78 readings and the subsequent inversion of the 12x12 matrix cannot feasibly be undertaken manually. It was therefore necessary to develop a computer program to reduce the amount of work involved. The program was required to take the raw data, with the minimum amount of alteration, and produce a full 18x18 stiffness matrix [K].

First of all the computer has to read-in the displacement measurements from one load case and find the unit force displacements. This was done by working out a least squares error straight line through each of the displacement measurements for the incremental forces. The gradient of the line is the unit force displacement. It is then necessary to transform the displacement measurements into the x,y,z,θx,θy,θz degrees of freedom at the two loading points. This is done from the geometry of the joint by:

1) Taking the average of the two displacements measured on the cross pieces as the deflection degree of freedom, and

2) The difference divided by the distance between the measuring points as the rotational degree of freedom. The matrix may now be formed into the true load cases by finding the two pairs of anti-symmetric forces and adding and subtracting the equivalent displacement rows. Where the necessary pair of load cases are not available it is sometimes required to subtract a previously calculated true load case from the mixed case. i.e. if the applied force vector was:

$$[1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$$

but there was no vector of the form:

$$[1\ 0\ 0\ {-1}\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$$

Then the true load cases:

$$[1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$$

and

$$[0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$$

have to be used to obtain the true vector. By simple re-arrangement it is now possible to produce a unit force matrix (diagonal terms=1) under which circumstances the displacement matrix is identical to the flexibility matrix:

$$[F][R]=[U], \quad [R]=[1] \quad \text{then} \quad [F]=[U]$$

The geometry of the joint is used to produce the equilibrium matrix from equation (12) and it is then a simple matter to calculate the full matrix from equation (11). A listing of this program and a flow chart are given in Appendix 8.

7.5 Initial results.

The box-sections had caused some problems in manufacture and had distorted a small amount, probably less than 1mm on the cross section. Since it was not known what effect this would have on measurements the experiment was conducted despite the imperfection. The maximum applied force of 100N produced displacements ranging from zero to 0.1mm. Some problems were encountered with the placement of the DTI's where the cross-pieces had become distorted due to welding by up to 5° from flatness. It was decided to set the gauges perpendicular to the surface so that no readings would be produced in one direction by displacements in the perpendicular direction. This should produce little error since $\cos 5^\circ = 1.0$ to less that 1/2%, thus the small change in direction of measurement should make an insignificant difference to the displacements measured.

The results produced under these conditions were digitised and read into the computer, at which point it was found that negative leading diagonal stiffness terms were produced in the final stiffness matrix. These are of course impossible, since all "direct" stiffness terms must be positive for the conservation of total energy. To find the reason for this, extra readings were taken to check the symmetry of the flexibility matrix. This highlighted some error in the direction of some measured deflections, but when corrected the negative diagonal terms still persisted. The symmetry terms were not always identical, some varying widely. To counteract this a routine was added to take the mean of any symmetrical terms. This again had no effect on the negative stiffness terms.

A second complete set of measurements was taken for comparison, and though most of the readings were similar some large discrepancies were noted, especially on the smaller readings. Again this new set of results produced negative diagonal stiffness terms. A combined average set of the two readings also made no improvement.

It was obvious from these results that considerable care was required to produce reliable answers.

7.6 Modifications to improve results.

Three main modifications were made, one to the joint, one to the test apparatus and one to the program.

1) Each cross piece on the joint was straightened to within 1° and where distortion was very bad the cross pieces were replaced.

2) The forces were now applied through a knife edge instead of the vertical bolt to ensure that no side forces could be induced by the bolt lying against the edge of the hole.

3) The program was modified to calculate a correlation coefficient for each of the least squares lines. It was to flag a correlation of below 0.999 and signal an error if below 0.995. This gave a very tight tolerance on the quality of the results.

Two more complete sets of readings were taken under these new conditions, but it was still not possible to obtain a stiffness matrix without negative leading diagonal terms. It was obvious that the somewhat unreliable techniques being used for the raw measurements were not able to cope with the complex situation. A major problem was the hysteresis in the structure. If it was disturbed slightly during a set of measurements all the readings could change by up to 10% of the maximum reading.

At this point it was decided that a considerably more sophisticated experimental procedure was going to be required, and with this in mind a decision was made to develop a laboratory computer based procedure to do the testing. Chapter 8 describes the microcomputer controlled system designed with the aim of overcoming all the above problems.

When it was not found possible to produce a valid stiffness matrix from the measured values for comparison with the finite element model, it was decided to compare the displacements instead. For the major displacements the magnitudes were similar but displayed a different stiffness distribution. The finite element model displayed higher bending stiffnesses and lower torsional stiffnesses than the measured joint. Differences were as large as 50% (considerably higher on many of the minor terms).

A check on the modelling accuracy of the joint was made by a comparison with the experimental beam measurements given in chapter 4. The beam was modelled with a similar mesh density to that of the joint, that is, one element across the flange and three across the plate. Again the torsional stiffnesses were low and bending stiffnesses high. It was not until the number of elements across the flanges was increased from 1 to 3 that the results came to within 5% of beam theory. Increasing the number of flange elements on the joint model would considerably increase the front-size and therefore the cost of the analysis. It is however, unlikely that reliable theoretical results can be produced without this refinement to the mesh on the flanges. It may, however, be possible to produce a special element which can represent the stresses around a spotweld on a flange, since the large stress gradients in such a situation are inadequately represented by a single element.

Another area for concern was the representation of the spotwelds themselves. These were simply taken as rigid connections between the two panels at the centre of the spotweld, by connecting the translatorary degrees of freedom at the two (or more) coincident nodes. To check that this was an acceptable approximation, two simple analyses were undertaken.

1) A spotweld on a plate was modelled using plane-stress elements (see Figure 7.7).

2) The model was then simplified so that there was just one rigid connection at the centre of the spotweld.

The analyses produced very similar patterns and levels of stress on the plate showing that that the simplified version should give a reasonable representation of a spotweld. The representation of the spotweld should therefore be adequate as it stands without any increase in sophistication.

7.7 Conclusions.

1) It is not possible to obtain an accurate set of joint stiffness/displacement measurements without considerable complexity. Due to the structural hysteresis it is necessary to repeat the measurements many times to obtain reliable results. With the number of measurements required this is impossible by hand.

2) In an analysis of a fabricated structure, the stress gradients in the flanges may be severe and therefore require a relatively refined mesh for reasonable accuracy.

3) Insufficient fineness of mesh density in this case produces high bending stiffnesses and low torsional stiffnesses.

Figure 7.1 Model of a Metro D-Post joint.

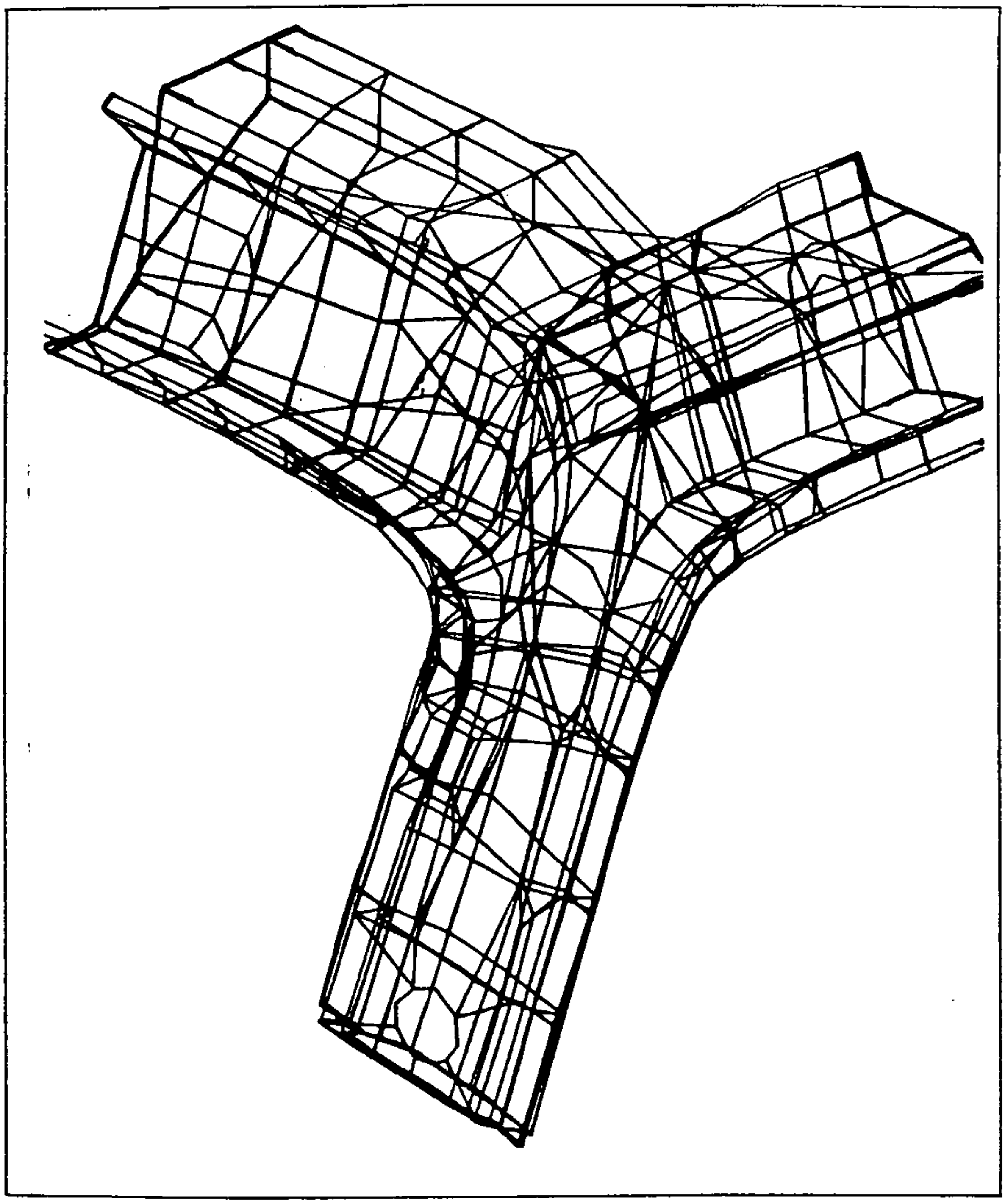


Figure 7.2 T-Joint model

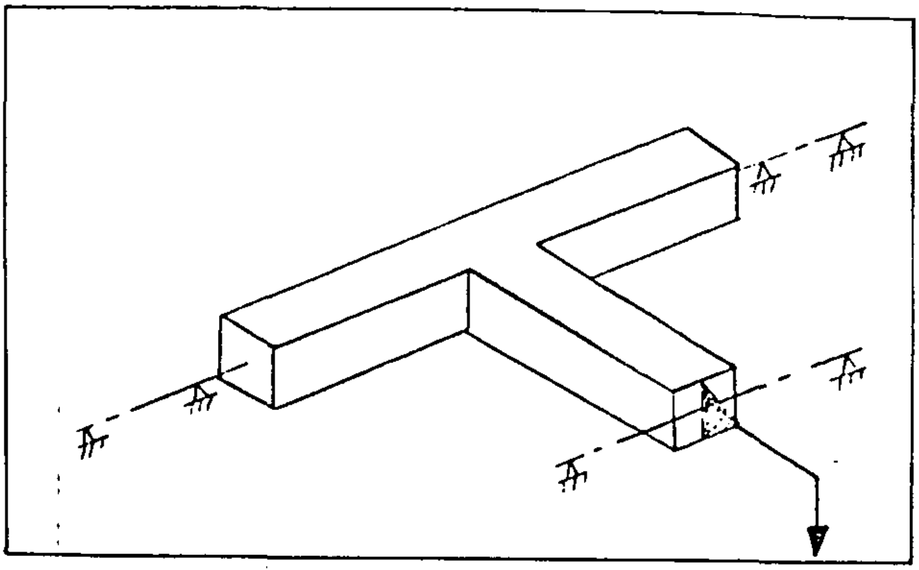


Figure 7.3 Joint representation by springs.

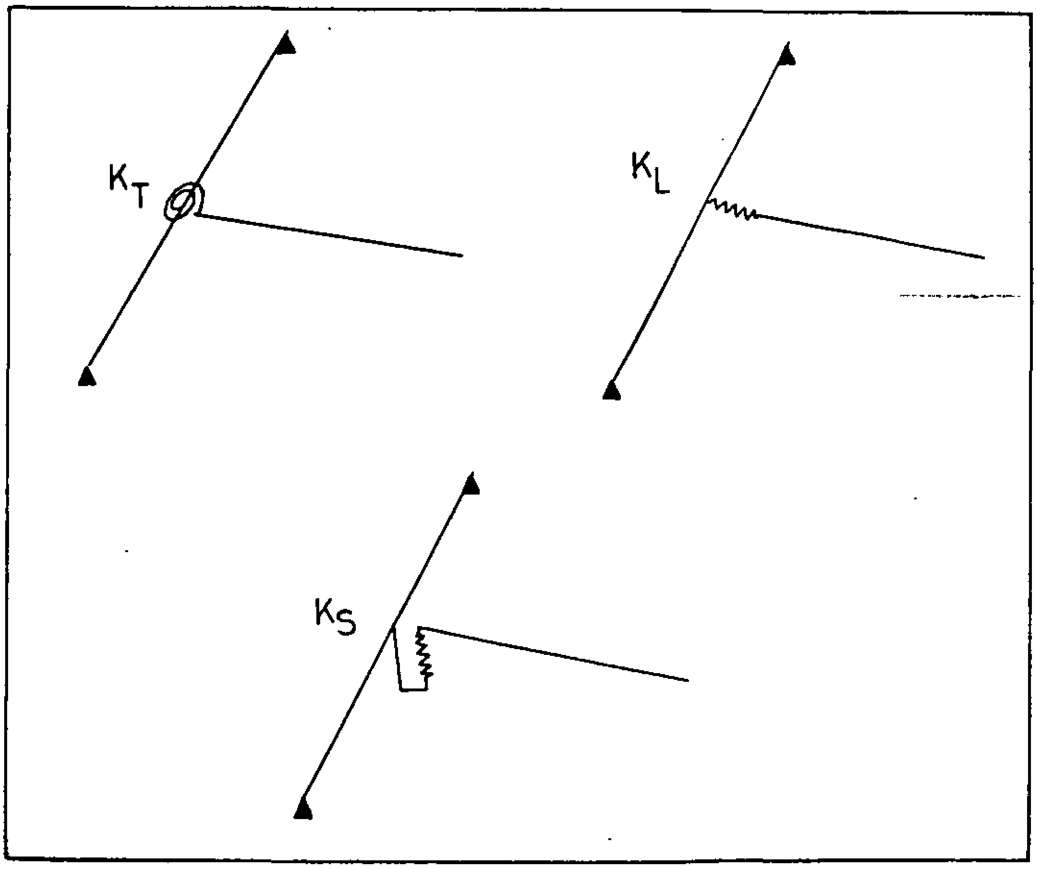
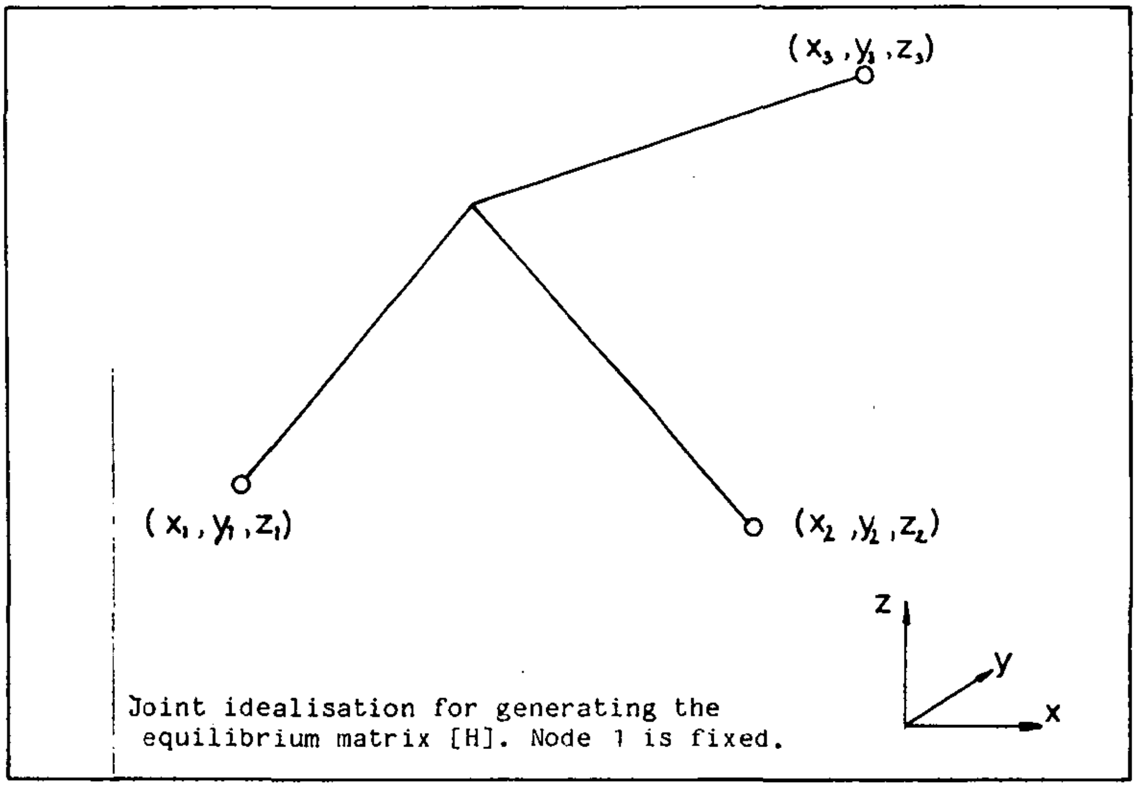


Figure 7.4 Idealised joint freedoms.



Joint idealisation for generating the equilibrium matrix [H]. Node 1 is fixed.

Figure 7.5 Equilibrium and load matrices.

$$
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & (z_2 - z_1) & (y_1 - y_2) & 1 & 0 & 0 & 0 & (z_3 - z_1) & (y_1 - y_3) & 1 & 0 & 0 \\
(z_1 - z_2) & 0 & (x_2 - x_1) & 0 & 1 & 0 & (z_1 - z_3) & 0 & (x_3 - x_1) & 0 & 1 & 0 \\
(x_1 - x_2) & (y_2 - y_1) & 0 & 0 & 0 & 1 & (x_1 - x_3) & (y_3 - y_1) & 0 & 0 & 0 & 1
\end{bmatrix}
\qquad
\begin{bmatrix}
X_2 \\ Y_2 \\ Z_2 \\ M_x \\ M_y \\ M_z \\ X_3 \\ Y_3 \\ Z_3 \\ M_x \\ M_y \\ M_z
\end{bmatrix}
$$

Figure 7.6 End diaphragm fittings.

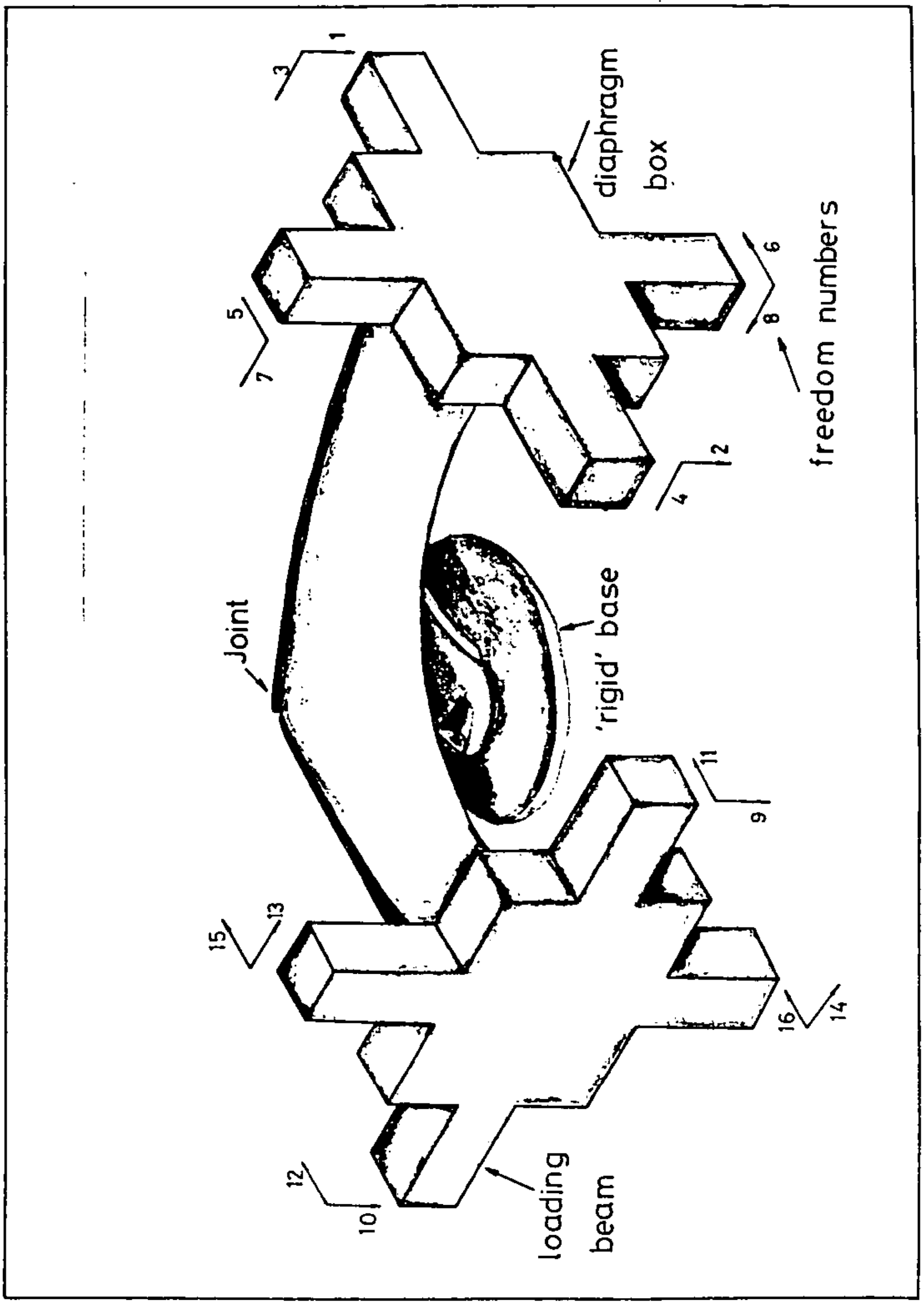


Figure 7.7 Spotweld model Stresses

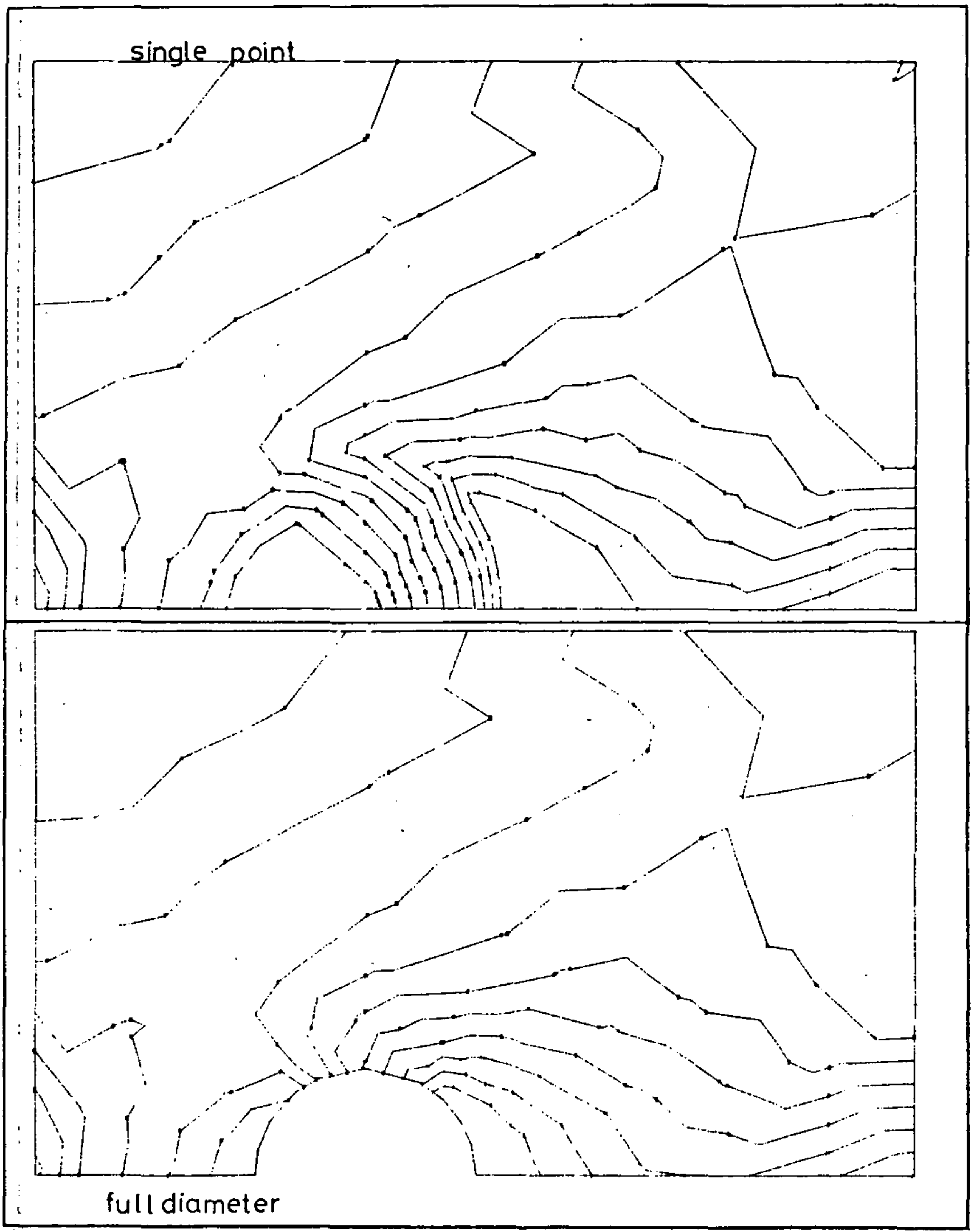
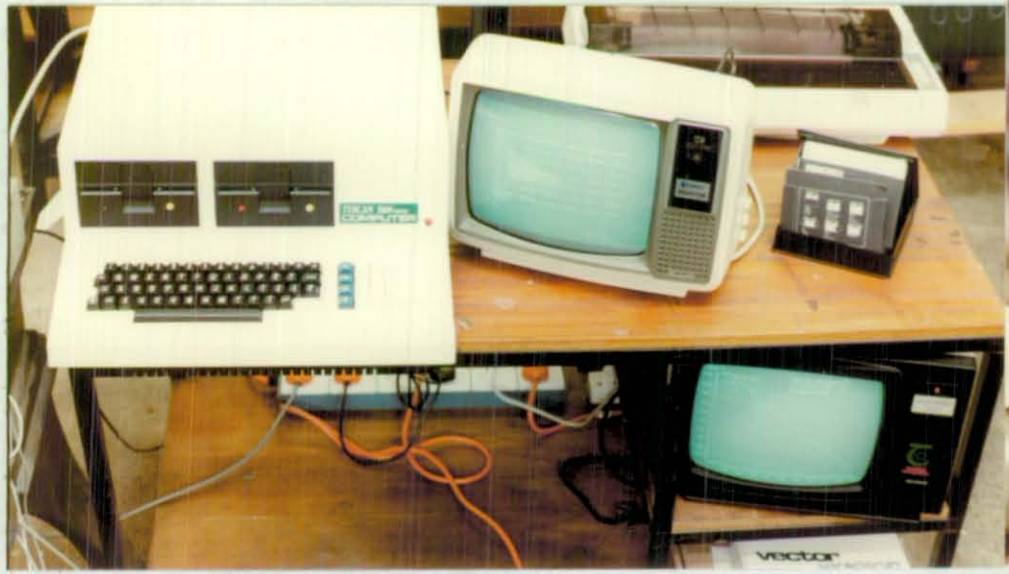
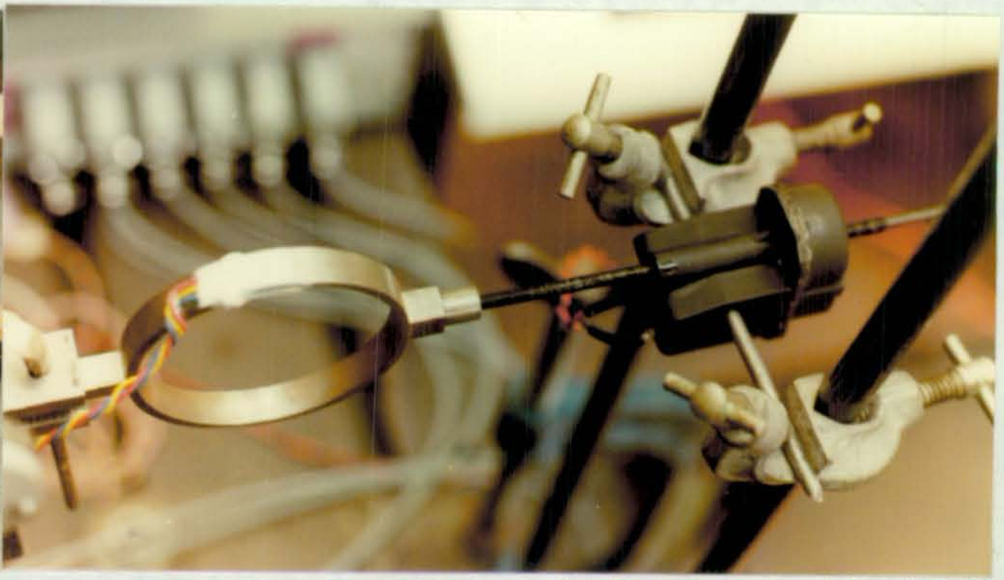


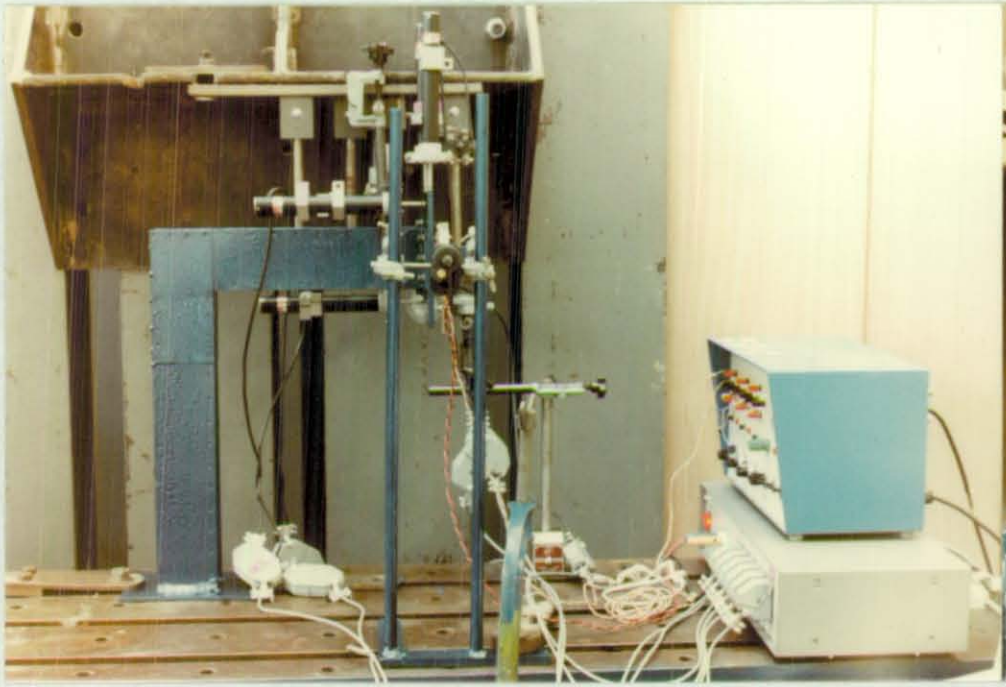
PLATE 4

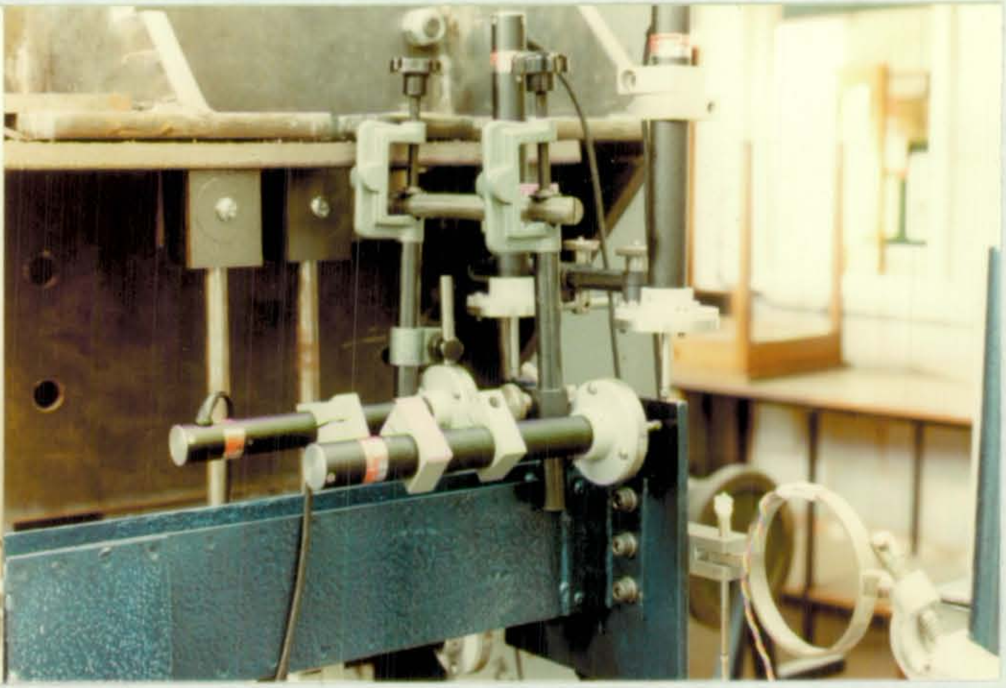




Top left: Micro-computer Layout
Bottom Left: Test Bench Layout

Top Right: Loading Motor and Load Cell
Bottom Right: Potentiometric Displacement Transducers





Chapter 8.

Laboratory Computer Data Logger.

8.1 Introduction.

The available equipment in the structures laboratory at Loughborough University was found to be inadequate for the measurement of the multi-point loads and deflections on the vehicle joint. To facilitate the more accurate measurement of the force and deflection parameters it became obvious that some automatic process was required since this was not feasable manually. The most efficient and readily available solution was the use of a micro-computer. This would then be able to run the experiment repeatedly, quickly and accurately and could be left unattended. The computer therefore had to fulfill the following requirements:

1) Apply and measure loads.
2) Measure deflections.
3) Display and analyse results.

For flexibility an S100 (IEEE Standard) bus system was chosen so that any additional computer boards would be easy to obtain. In its standard form the computer is unable to take any analogue measurements from the outside environment and thus requires an analogue to digital converter which is capable of converting the analogue load and displacement signals to digital form. Both sets of readings can be taken with the same ADC and an accuracy of >1% on each reading would be considered adequate. To obtain this level of accuracy a 12-bit ADC is required (this can read to a precision of one part in 4096) with an accuracy of +/- 2 giving a resolution of 0.1%. ADC's can only read over a specific voltage range (say +/- 10Volts) therefore if a potentiometer displacement transducer with a stroke of 25mm is used with -10V on one end of the rheostat and +10V on the other, the ADC will only be able to read to an accuracy of 0.01mm which is not adequate for the small deflections expected. Even with programmable gain ADC to increase precision it would not be possible to gain much accuracy exept at the mid-span of potentiometer about the zero voltage level. Any other part of the scale would be above the full scale defelection of the ADC. To overcome this problem it was decided to use a DAC to set the end voltages on the potentiometers thus allowing the datum position of the potentiometers to be set at zero volts. This allows the ADC gain to be set to give a full scale deflection (FSD) of similar magnitude to the measured deflection, thus keeping the accuracy of the readings close to the resolution accuracy of the ADC.

This then specifies the requirements of the displacement transducers and two computer interface boards. The transducers chosen were Penny and Giles 25mm Hybrid Track Potentiometers which have almost infinite resolution and a linearity of better than 0.03%. The ADC chosen was a 12-bit device manufactured by CDC in America. It has a multiplexed analogue input of 8-differential or 16 single-ended channels. The gain on each channel can be set directly by the computer to give FSD's of from +/- 10V down to +/- 10mV. The DAC was simply required to apply an accurate and stable voltage for the potentiometers and an 8-bit accuracy was considered adequate for this.

The application and measurement of force was all that now remained. The force is most easily applied by an electric motor with a linear actuator mechanism and is easily controlled by the computer. Electric Linear actuators are expensive for such a specialized application but a low cost device maufactured by Smiths Industries was found to be adequate, if a little low on load capacity. The motor was controlled by a 2-bit code from a parallel port on the computer. 11 or 00 = off, 10 = pull, 01 = push.

The measurement of the load applied by the motor required a load cell with good sensitivity to the relatively small forces. For this purpose a strain-gauged ring was designed. To give the required sensitivity the maximum applied load should give a strain of about 500 microstrain. The bending moment developed in a ring is:

$$M_o = P\ R(\tfrac{1}{2} - \frac{1}{\pi})$$

across the centre. If R (the radius of the ring) is chosen as 40mm and the load is 100N, then the bending moment produced is .73Nm. If the ring is 10mm deep and y mm thick it will have a second moment of area of:

$$I = \frac{bd^3}{12} = \frac{.01 \times y^3}{12} = 833 \times 10^{-6}\ y^3 \quad m^4$$

To achieve a strain of 500 microstrain we require a stress of:

$$\sigma = \varepsilon E = 500 \times 10^{-6} \times 200 \times 10^9 = 100 \times 10^6\ N/m^2$$

which will be a combination of the bending and direct stress in the ring, thus:

$$\sigma = \frac{M.y/2}{I} + \frac{P/2}{.01y} = \frac{.73/2}{833 \times 10^{-6} y^2} + \frac{100/2}{.01y} = 100 \times 10^6$$

gives y as 2.12mm. For safety a thickness of 2.5mm was chosen to avoid overstressing. This ring was fitted with a half bridge strain gauge and the output signal was fed via a strain gauge amplifier to the ADC. The calibration curve for the load cell is shown in Figure 8.2.

The layout of the hardware is shown in Figure 8.1.

Control of the apparatus was performed by a program written in FORTRAN 80 with a few machine code routines. Control of the individual devices is described below.

8.2 Operation of the ADC.

The ADC is controlled through 4 input/output ports, with a base address at A0h. Each port controls a different function of the ADC.

Output A0h	Sets the gain and operation mode. (This is usually the WAIT until ready mode)
Output A1h	Sets the channel number to be read.
Output A2h	Causes an A to D conversion to be made.
Input A2h	Read Low byte of digitised signal.
Input A3h	Read High byte (2's complement).

This procedure is carried out by a machine code subroutine by using CALL ADREAD(ICHAN,IGAIN,IVALUE) from a FORTRAN program. The gain is set on two amplifiers each with a two bit gain code. To convert the actual gain required to the four bit code the subroutine CALL GAINAD(IGREQ,IGACT,IGAIN) converts the gain required IGREQ, to the next lowest gain available from the ADC, IGACT is then the actual gain value set and IGAIN is the four bit code to send down in ADREAD.

8.3 Operating the DAC board.

The DAC board required for the experimental work was a 20V FSD device with a 10mA output current, manufactured by Transam Computers. Unfortunately this was not ready in time and a Cromemco 7A+D board had to be used instead. This board is only a +/- 5V device and has an output current of only 1.5mA, which is not really adequate. In use the 1.5mA proved optimistic since the output was not stable with any current drain greater than 0.5mA. This meant that the ADC was always working at maximum gain where its accuracy is severely reduced.

The DAC is much easier to operate than the ADC and has one port controlling each channel. The base address is 18h which is the parallel input/output port. The next 7 addresses are the analogue input/output channels. To simplify the calling sequence a FORTRAN routine CALL DAC(VOLTS,ICHAN) was written where VOLTS is the required REAL*4 voltage and is returned as the nearest voltage which can be set ($-2.56<VOLTS<2.56$) and ICHAN is the channel number (from 1-7) for the voltage to be set on.

8.4 Reading the voltages off the potentiometers.

Initially it had been intended to set a potential of 10V across each pair of potentiometers to give a high voltage gradient without taking too much current from the DAC's. (The potentiometers have a resistance of 1kΩ each, thus a pair in series would draw 5mA at 10V). Unfortunately the use of the Cromemco board required all the potentiometers to be put in series with a potential of only 2V. This gives only 0.33V across each potentiometer instead of the intended 5V. This results in a large amount of noise being superimposed on the signal since the ADC is reading smaller voltage changes on the potentiometers than the accuracy that the DAC can be set to. The only way to overcome this problem without the better board was to take a large number of readings and average them.

The object of using the DAC's to set the potentiometer voltages is to keep the voltage swings produced by the deflections just inside the FSD of the ADC thus using the full precision of the ADC. To find the zero datums a 0-2V potential was set across the potentiometers and their datum positions read. After applying the maximum load the wipers were re-read and the voltage change noted. The computer can then calculate the required end voltages to make the datum approximately zero volts and the gain for FSD at maximum load. With the voltages and gain known it is now possible to calculate the deflection at any point between the zero and maximum load datums.

8.5 Experiment program.

A listing of the FORTRAN program used to run the joint measurements experiment is given in Appendix 9. It is divided into 3 sections: (i) Calibration, (ii) Acquisition of results and (iii) Check on accuracy.

The calibration section first requires to know the maximum load which the motor has to apply. The motor used has a maximum load capability of 67N, any greater load will damage the gear train. It is possible to modify a plasic spigot in the gear train to increase the load capability. With a brass spigot the motor is capable of applying a load of 120N - though now causing more strain on the motor itself. The unmodified motor showed some signs of wear when it was repeatedly run to its maximum load so that the modified motor was only run to 70N for the sake of reliability.

The program next measures the datum zero load, any change from this initial reading is then proportional to the applied load. This method can cause problems if the load is high when the program starts to run, since damage may be done to the motor if it is overloaded. To overcome this, the routine zload (zero load) is used to run the motor in reverse until the load cell is slack.

The program can now obtain the datum values for the displacement potentiometers. A voltage is set across all the potentiometers and each wiper is read in turn. This allows the computer to calculate the end voltages which need to be set to give approximately zero volts at the datum zero wiper position. The maximum load is then applied and the wipers are re-read. The difference in the two readings allows the gain for the ADC's to be calculated to give a FSD. Since the DAC is only 8-bit it is possible that the DAC's cannot set the required voltages for the ADC at full gain. The gain is therefore limited to 512. A routine is used to fine tune the voltages when the precise required voltage cannot be achieved. If this routine fails to find a suitable voltage it requests operator intervention.

To avoid hysteresis problems the load zero datum is retaken after the calibration load cycle.

The computer is now ready to start the actual stiffness measurements. All it needs to know is the number of increments the load is to be divided into and the number of readings to take at each load. More than one reading must be taken as there is noise on the D.C. signals (at 4MHz, the computer clock frequency) causing considerable variability. Though 10 averaged readings sometimes gives consistent results, 100 readings generally gives better reliability.

The computer calculates the required load and drives the motor to set that load using the routine: CALL SETLD(REQLD,DATLD) The routine switches on the motor and continually reads chanel 7 of the ADC until the required load is reached. Since the dynamic load is higher than the static load the reading has to be rechecked when the motor is switched off. If the load is too low, the motor is switched on again and the loading loop is repeated until a satisfactory load is reached. This sometimes causes some stuttering of the motor as it is switched on and off rapidly.

The routine CALL DISP now takes the required number of readings and averages them. The load is incremented and all the readings re-taken until the maximum load is reached. The routine CALL LSTSQ now takes all the points for each potentiometer and finds the best straight line to fit through them. The correlation coefficient is also calculated to give a measure of the repeatability of the results. The plotting routines now present the results graphically for the operator to view. If they are unsatisfactory the program can be restarted with a different number of averaged readings.

If the results are all acceptable the program then repeats the load cycle until the average gradients calculated for each potentiometer settle to within 1/2% of the previous average.

Figure 8.3 shows the output from a typical program run.

8.6 Results.

It had been intended to compare the experimental results produced here with those from a finite element model. A model was developed using a similar mesh refinement to that found necessary for the analysis of the beams described in Chapter 4. In fact the experimental joint used for this work was made from two of these beams, joined together in a way similar to modern automotive practice. Figure 8.4 shows a drawing of the model - no symmetry constraints were employed since the front would not exceed 150 (the elements used -36210- only have 2 degrees of freedom per node).

When run this model used the maximum allowed CPU time of 546 mins. without completing the setting up of the solution (Phase 4). It was therefore not possible to obtain any results. The model may have been tractable with symmetry taken into account, but it must be noted that any practical automotive joint will be far more complicated than this one, and will not have any symmetries. It is therefore unlikely that an accurate model could be developed, with the necessary three elements across the flanges, which could be analised at a resonable cost on the computer facilities at Loughborough.

The measured deflections are, as expected, similar in magnitude to those produced in the beam experiments of chapter 4. It is, however, noticable that some deflections show a considerable drift before settling to a stable value. It would not be concievable to undertake this many readings by hand.

Since there are no figures to compare these results to, they are not presented here as they are meaningless on their own.

8.7 Conclusions.

1) The computer is capable of producing a set of results much more rapidly than can be achieved manually. The reliability of the results also appears to be better. There are changes in measured stiffness during a run, but nothing as drastic as sometimes found during manual readings when jogging the apperatus could completely change the results. There is usually no need to approach the apparatus when the computer is in control, so that no disturbances should be caused.

2) The facility to plot data immediately makes the acquisition of reliable results much quicker and simpler as erroneous results can be spotted quickly.

3) The computer is capable of displaying results in their final format while the experiment is still, underway thus making it easier to check their validity.

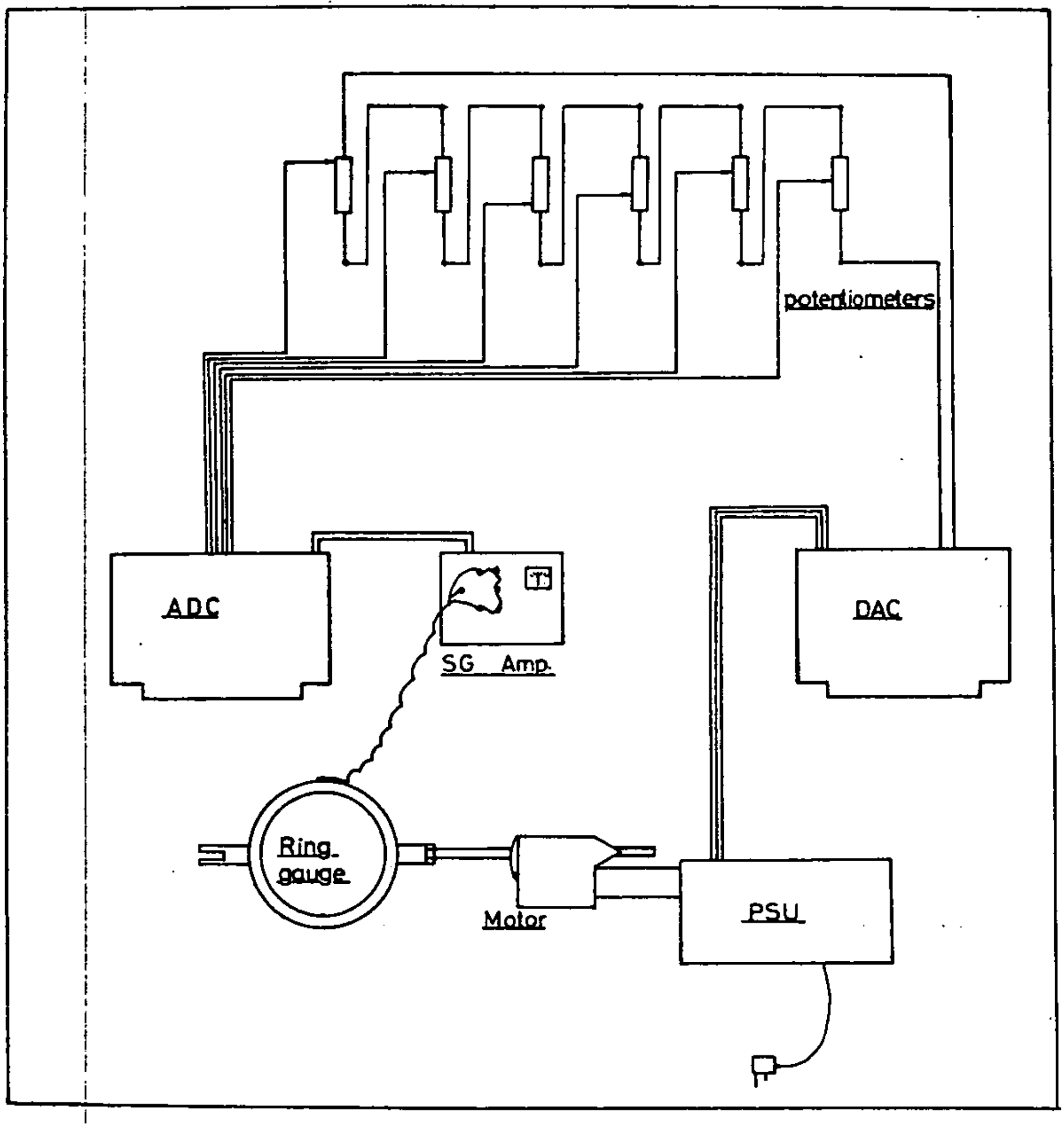


Figure 8.1 Schematic computer layout.

Figure 8.2 Calibration curve for load cell.

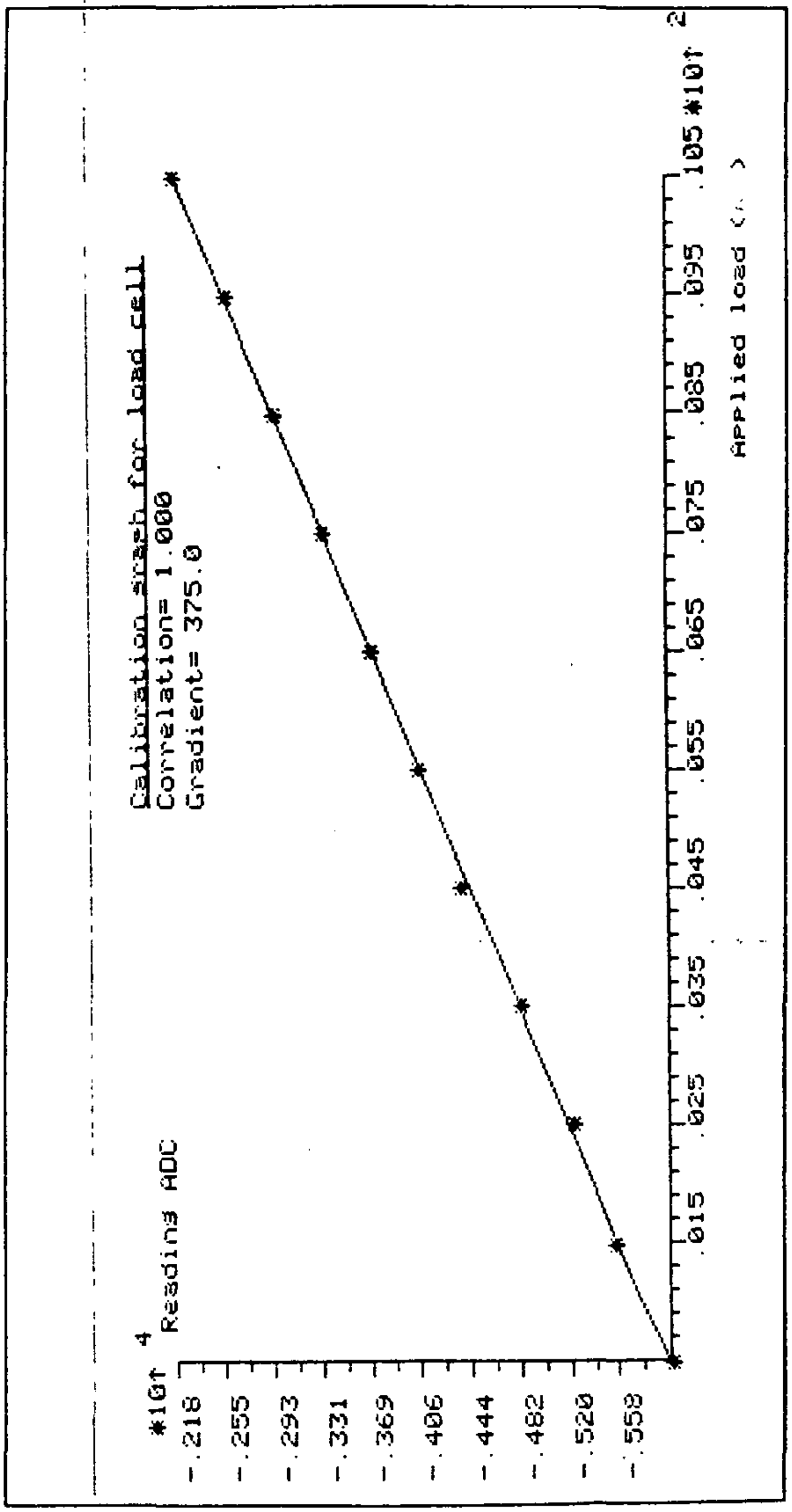
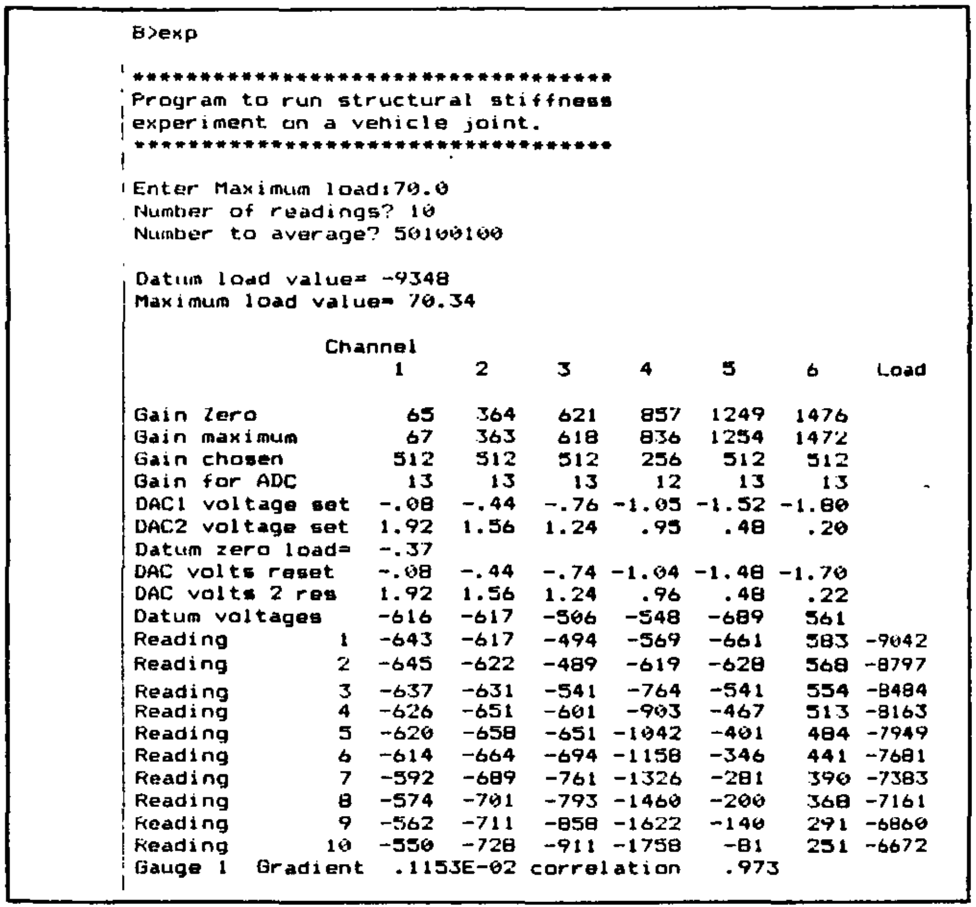
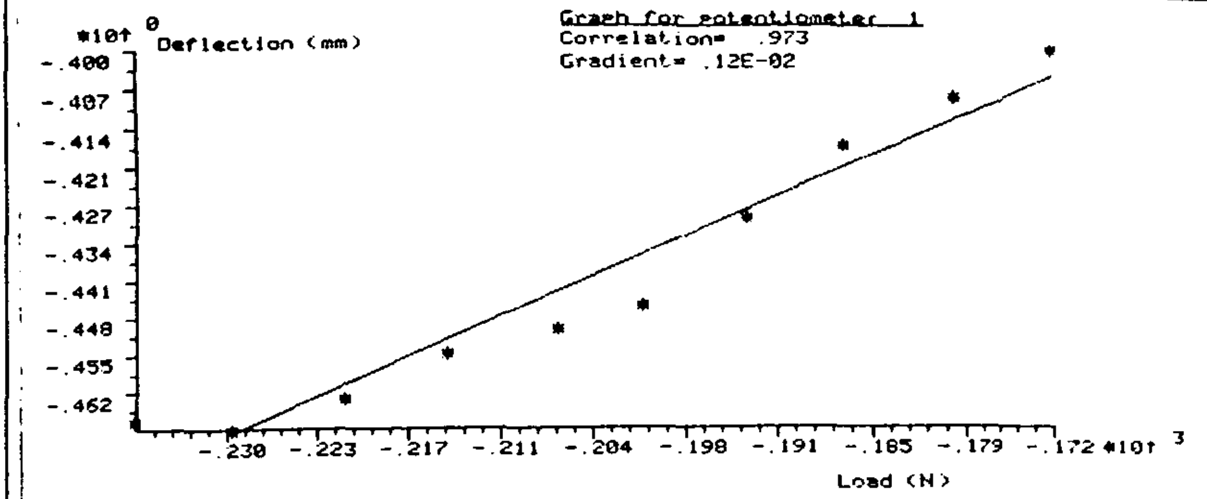
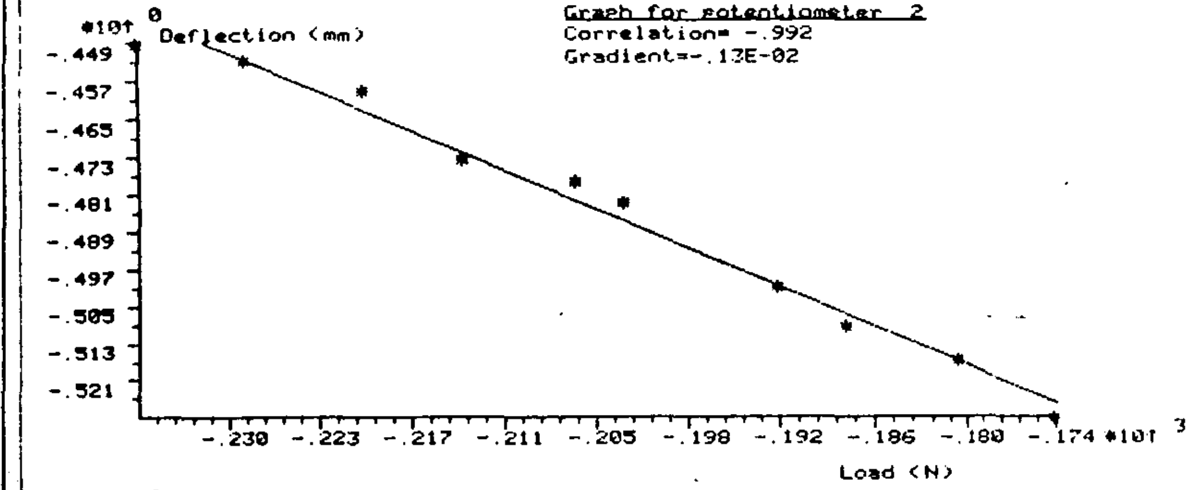


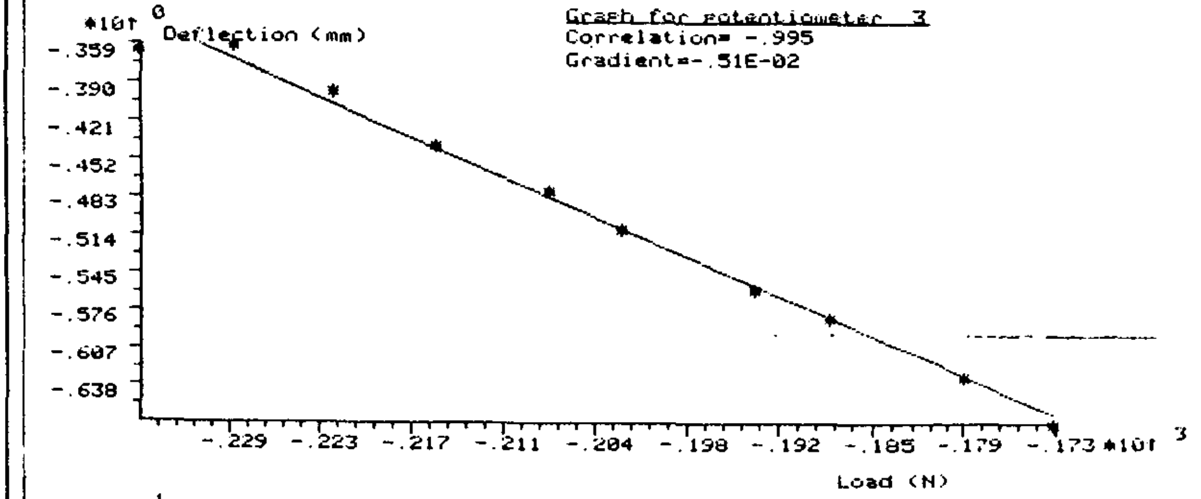
Figure 8.3 Typical program run.

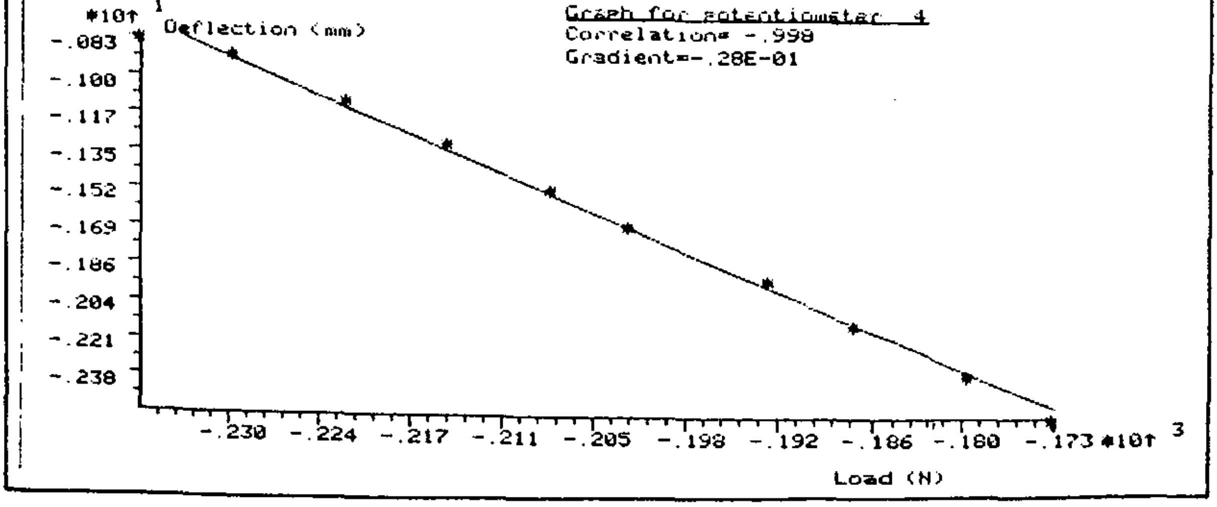


```
B>exp

***********************************
Program to run structural stiffness
experiment on a vehicle joint.
***********************************

Enter Maximum load:70.0
Number of readings? 10
Number to average? 50100100

Datum load value= -9348
Maximum load value= 70.34

              Channel
                   1     2     3     4     5     6   Load

Gain Zero         65   364   621   857  1249  1476
Gain maximum      67   363   618   836  1254  1472
Gain chosen      512   512   512   256   512   512
Gain for ADC      13    13    13    12    13    13
DAC1 voltage set -.08  -.44  -.76 -1.05 -1.52 -1.80
DAC2 voltage set 1.92  1.56  1.24   .95   .48   .20
Datum zero load= -.37
DAC volts reset  -.08  -.44  -.74 -1.04 -1.48 -1.70
DAC volts 2 res  1.92  1.56  1.24   .96   .48   .22
Datum voltages   -616  -617  -506  -548  -689   561
Reading        1 -643  -617  -494  -569  -661   583 -9042
Reading        2 -645  -622  -489  -619  -628   568 -8797
Reading        3 -637  -631  -541  -764  -541   554 -8484
Reading        4 -626  -651  -601  -903  -467   513 -8163
Reading        5 -620  -658  -651 -1042  -401   484 -7949
Reading        6 -614  -664  -694 -1158  -346   441 -7681
Reading        7 -592  -689  -761 -1326  -281   390 -7383
Reading        8 -574  -701  -793 -1460  -200   368 -7161
Reading        9 -562  -711  -858 -1622  -140   291 -6860
Reading       10 -550  -728  -911 -1758   -81   251 -6672
Gauge 1  Gradient  .1153E-02 correlation   .973
```

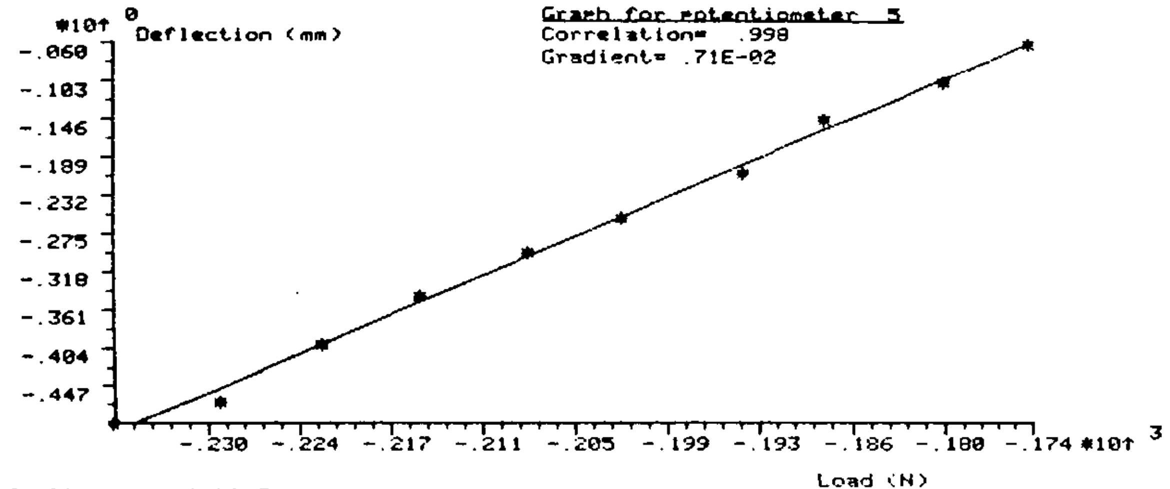

Graph for potentiometer 1
Correlation= .973
Gradient= .12E-02
*10↑ 0
Deflection (mm)
-.400
-.407
-.414
-.421
-.427
-.434
-.441
-.448
-.455
-.462
-.230
-.223
-.217
-.211
-.204
-.198
-.191
-.185
-.179
-.172
*10↑ 3
Load (N)


Graph for potentiometer 2
Correlation= -.992
Gradient=-.13E-02
*10↑ 0
Deflection (mm)
-.449
-.457
-.465
-.473
-.481
-.489
-.497
-.505
-.513
-.521
-.230
-.223
-.217
-.211
-.205
-.198
-.192
-.186
-.180
-.174
*10↑ 3
Load (N)

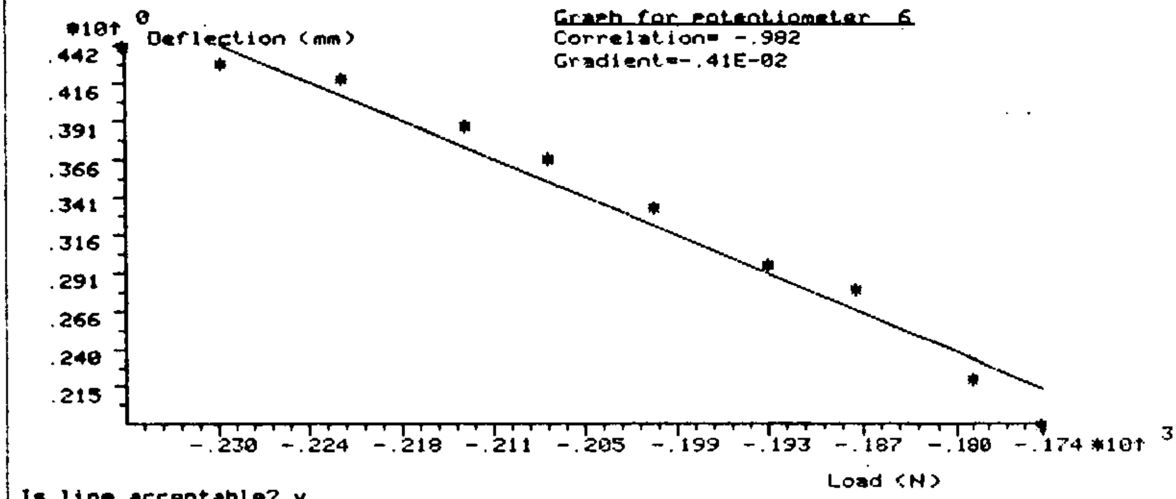

Graph for potentiometer 3
Correlation= -.995
Gradient=-.51E-02
*10↑ 0
Deflection (mm)
-.359
-.390
-.421
-.452
-.483
-.514
-.545
-.576
-.607
-.638
-.229
-.223
-.217
-.211
-.204
-.198
-.192
-.185
-.179
-.173
*10↑ 3
Load (N)


Graph for potentiometer 4
Correlation= -.998
Gradient=-.28E-01
*10↑ 1
Deflection (mm)
-.083
-.100
-.117
-.135
-.152
-.169
-.186
-.204
-.221
-.238
-.230
-.224
-.217
-.211
-.205
-.198
-.192
-.186
-.180
-.173
*10↑ 3
Load (N)

```
Print graph?y
```

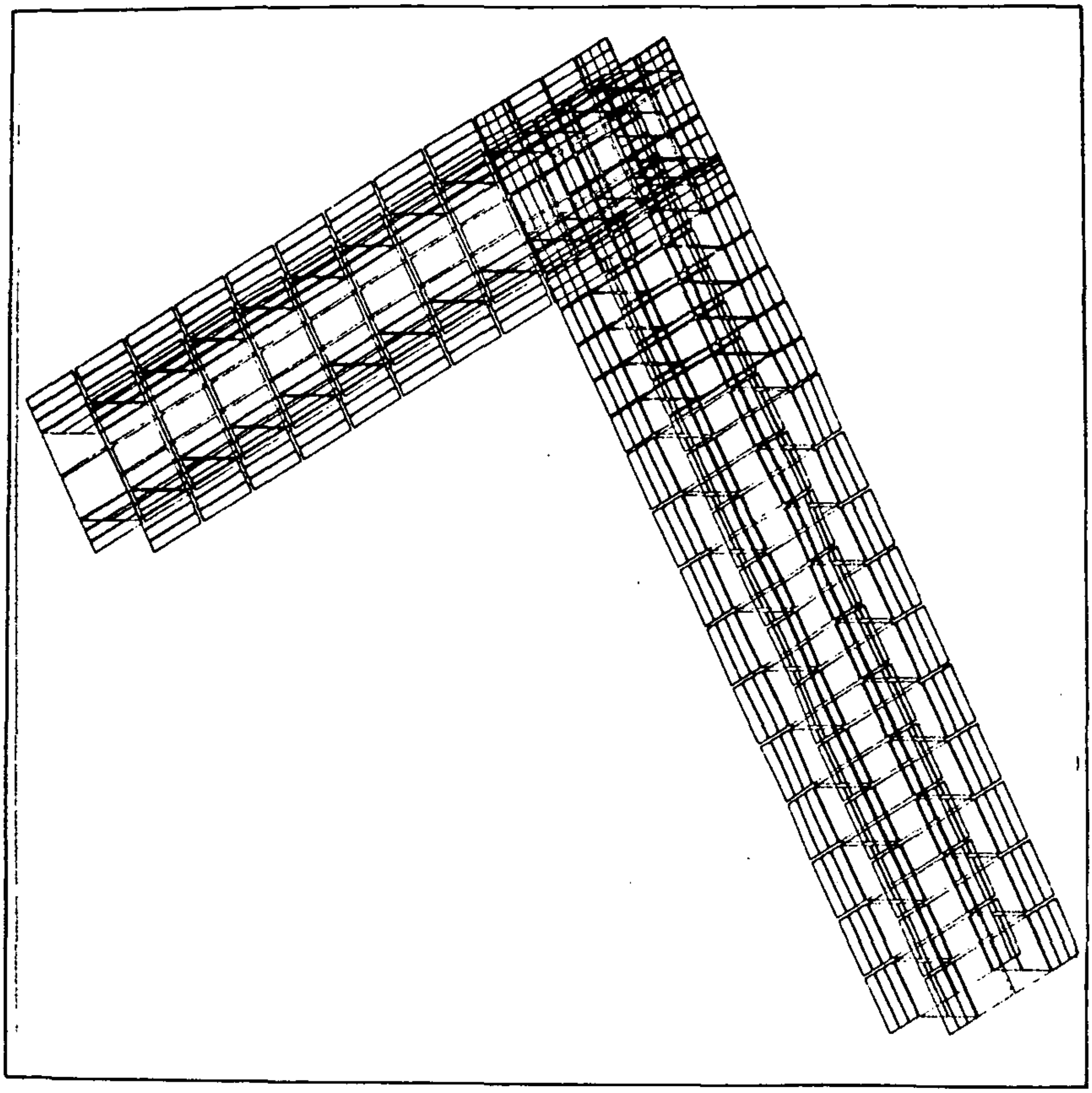


```
Is line acceptable? y

Gauge 6  Gradient -.4094E-02 correlation  -.982

Print graph?y
```



```
Is line acceptable? y


Gradients=   .115E-02 -.135E-02 -.515E-02 -.282E-01  .712E-02 -.409E-02
Gradients=   .115E-02 -.141E-02 -.530E-02 -.287E-01  .693E-02 -.433E-02
Gradients=   .111E-02 -.142E-02 -.529E-02 -.288E-01  .691E-02 -.435E-02
Gradients=   .108E-02 -.144E-02 -.527E-02 -.286E-01  .684E-02 -.437E-02
Gradients=   .107E-02 -.147E-02 -.524E-02 -.284E-01  .680E-02 -.440E-02
Gradients=   .107E-02 -.149E-02 -.523E-02 -.284E-01  .680E-02 -.439E-02
Gradients=   .106E-02 -.150E-02 -.521E-02 -.283E-01  .681E-02 -.438E-02
Gradients=   .105E-02 -.152E-02 -.518E-02 -.282E-01  .680E-02 -.437E-02
Gradients=   .104E-02 -.154E-02 -.519E-02 -.282E-01  .679E-02 -.436E-02
Gradients=   .103E-02 -.156E-02 -.519E-02 -.282E-01  .679E-02 -.436E-02
Gradients=   .102E-02 -.158E-02 -.519E-02 -.281E-01  .678E-02 -.436E-02
Gradients=   .101E-02 -.159E-02 -.517E-02 -.280E-01  .676E-02 -.435E-02
Gradients=   .101E-02 -.160E-02 -.517E-02 -.280E-01  .678E-02 -.434E-02
Gradients=   .998E-03 -.160E-02 -.517E-02 -.280E-01  .680E-02 -.434E-02
Gradients=   .992E-03 -.161E-02 -.517E-02 -.280E-01  .681E-02 -.433E-02
Gradients=   .991E-03 -.162E-02 -.517E-02 -.279E-01  .681E-02 -.433E-02
```

Figure 8.4 Finite element model picture.



OVERALL CONCLUSIONS

9.1) The work described in Chapter 2 was mainly an introduction for the Author to the finite element method but did produce some useful results and pointers for future work. The analysis gives some feel for the number of elements required in different situations to obtain reliable answers. The meshes described were very simplistic but were as advanced as the available computer facilities could cope with at the time.

9.2) The parametric analysis of vehicle components described in Chapter 3 produced many interesting findings. The windscreen pillar was found to be fairly ineffective in torsion even though it would be expected to form a part of a torsion box with the roof and floor. The roof was also not as effective in torsion as would be expected. This may be due to the large size of the peripheral roof beams.

In hindsight the complete removal of beams for this analysis seems to be too drastic, a change in properties of 10-20% may have proved more informative.

The analysis does point out a few areas which have little stiffening effect on the overall structure and could therefore be replaced with lightweight materials.

9.3) The measurements of beam stiffnesses agree well with previous workers for the spotwelded sections. There is little to be gained in bending stiffness by using adhesives since spotwelded beams achieve 93-98% efficiency, adhesive beams produce very similar values.

In torsion there is much more room for improvement and the efficiency is improved from 86-87% for spotwelds to 93-96% for adhesives. An extra stiffening effect is also present when using adhesives as the shear flow path is shortened, increasing the value of J from that of an identically sized spotwelded beam.

9.4) The analyses of adhesive joints in Chapter 5 show the care required in the choice of adhesive. A flexible adhesive layer will be evenly stressed but will reduce the efficiency of the overall structure while a stiff adhesive layer tends to have high stress concentrations which will damage the bond and cause failure of the structure. A simple analysis shows that using a typical epoxy resin to fabricate a box beam, its bending stiffness will be equal to that of a similar homogeneous beam.

9.5) Chapter 6 shows various ways of accounting for joint seams in beams when in torsion. The Batho-shear equations can be used to obtain approximate results. For more accurate results the finite element method can be used to solve St. Venents torsion equations. These can then accurately represent the adhesive joint layers. It is interesting to note from the results, the relatively large stiffening effect of the flanges, despite their small cross-sectional area.

9.6) The analysis and measurement of the joints was unfortunately not very conclusive. When trying to develop a full stiffness matrix it was not found possible to erradicate the negative leading diagonal terms, produced by inaccuracies in the measurements. The joint was subject to the non-linear effects and any disturbance during a measurement sequence would alter the datum.

The technique for producing the full stiffness matrix has been developed but further work would be worthwhile on the development of the measurement system.

The Micro-computer is able to repeat measurement sequences many times while left unattended and is therefore likely to produce more reliable results. Again this system has been fully developed, but the lack of time precluded its use on a real vehicle joint.

REFERENCES.

Effects of joints.

1) Chang D.C. 'Effects of flexible connections on body structural response.' SAE740041 1974.

2) Shigeta K.
Nakajima K.
Tanabe M.
'An analysis of a car body side structure with elastic joints.' J.SAE Japan Vol.20,No.2 1966

3) Sharman P.W. 'Some aspects of the structural analysis and design of commercial vehicles.' PhD.Thesis, Loughborough 1974

4) Sharman P.W. 'The effect of practical joints on the stiffness of framed structures.' TT7103 Dept.of Trans. Tech. Loughborough University

5) Sharman P.W. 'The effect of joint flexibility on the torsion of a vehicle body.' IMechE 15/82

Adhesives.

6) Adams R.D.
Coppendale J.
Peppiatt N.A.
'Failure analysis of Aluminium-Aluminium bonded joints.'

7) Adams R.D. 'Strength of CFRP lap joints.' Adhesion-4 Applied sci. pub.

8) Adams R.D.
Coppendale J.
'The stress strain behaviour of axially loaded butt joints.' J.Adhesion 1079 Vol.10,pp49-62

References.

9)	Adams R.D. Coppendale J.	'Measuremnt of the elastic moduli of structural adhesives by the resonant bar technoque.' J.Mech.Eng.Sci.Vol.18 No.3 1976
10)	Adams R.D. Peppiatt N.A.	'Stress analysis of adhesive bonded lap joints.' J.Strain.Anal.Vol.9 No.3 1974
11)	Lees W.A.	'New adhesives and the structural engineer.' Auto.Eng. Feb. 1982
12)	Potts D.	'New tests strengthen the case for adhesives.' Eureka Jan 1981

Torsion.

13)	Zienkiewicz O.C. Cheung Y.K.	'Finite elements in the solution of field problems.' The Engineer pp.507 Sept.24, 1965
14)	Herrman L.R.	'Elastic Torsional analysis of Irregular shapes.' J.Eng.Mech.Div.Proc.ASCE
15)	Zienkiewicz O.C.	'The Finite Element Method.' McGraw-Hill 1977
16)	PAFEC 75	Theory and results (manual)
17)	PAFEC 75	Systems (manual)
18)	Timoshenko S.P. Goodier J.N.	'Theory of Elasticity' McGraw-Hill 1970
19)	Ely J.F. Zienkiewicz O.C.	'Torsion of compound bars-a relaxation solution.' Int.J.Mech.Sci. 1960 Vol.1 pp.356-365

20) Muskhelishvilli N.I. 'Some basic problems of Mathematical theory of Elasticity.' Noordhoff, Greningen 1953

21) Filonenko-borodich. 'Theory of Elasticity.' Foreign Languages Publishing house, Moscow.

Weight reduction.

22) Chisholm C.J., Goff R.W. 'Dynamic torsion tests of a car body.' MIRA 1965/9

23) Honda N. 'The vehicle weight reduction by the finite element method.' FISITA Int.Cong.XVIII 80.3.2.1

24) Ludtke N.F., Kaminski H.L., Osen W.R. 'A study of the costs of a production deck lid for steel, Aluminium and Sheet Molding compound.' SAE 810785

25) Chang D.C., Kuang-ming Wu, Vella J.R. 'Potential mass reduction and Material cost penalties of body panels with alternate materials' SAE810229

26) Kennedy F.E., Hooven F.J. 'An analysis of automobile weight and its reduction by Aluminium substitution.' SAE770805

27) Kennedy F.E., Hooven F.J. 'Weight concious automobile design.' SAE77080

References.

28) Revellino M., D'Aprile F. 'Advanced composite materials for structural components.' FISITA Int.Cong. 80.3.2.4

29) Rolf R.L., Sharp M.L., Herbein W.C. 'Minimising the weight of Aluminium body panels.' SAE790164

30) Osborn J., Theil L. 'Optimal design of minimum weight structures.' SAE810686

31) Herbein W.C., Wolff N.P. 'Minimising the weight and cost of an aluminium deck lid.' SAE810783

32) Levy B.S., Preban A.G. 'The use of increased curvature to reduce the weight of body panels.'

33) Loasby K.M. 'The use and manipulation of aluminium in Aston Martin Lagonda cars.' SAE780177

34) Eissinger R.C., Jewell N.H., Livermore J.C. 'Development and evaluation of Aluminium body sheet panels.' Ford Motor Company

35) Rolf R.L., Sharp M.L., Stroebel H.H. 'Structural Characteristics of aluminium body sheet.' SAE770200

36) Motwani M.B. 'Application of Aluminium in body weight reduction.' SAE770306

General Bibliography.

37) Ali R. Hedges J.L. Mills B.	'Computer aided design applied to a model of a chassis type structure using the finite element technique.' IMechE AD PI/70 proc. Vol.184
38) Przemieniecki J.S.	'Theory of Matrix Structural Analysis.' McGraw Hill N.Y. 1968
39) Bathe K.J. Wilson E.L.	'Numerical methods in Finite Element Analysis' Prentice Hall
40) Livesley R.K.	'Matrix methods in Structural analysis' Pergamon press 1964
41) Azar J.J.	'Matrix structural analysis.' Pergamon press 1972
42) Rockey K.C. Evans H.R. Griffiths D.W. Nethercot D.A.	'The Finite Element Method - A Basic Introduction for Engineers' Granada publishing N.Y. 1975
43) Kuske A. Robertson G.	'Photo Elastic Stress Analysis' John Wiley and Sons 1974
44) Hearn E.J.	'Photoelasticity' Merrow Technical Library Practical Series
45) Barrass R.	'Scientist Must Write' Chapman and Hall 1978
46) Coleman P. Bramkleby K.	'The Technologist as Writer' The Ryerson Press 1969
47) Microsoft	FORTRAN 80 Manual
48) Microsoft	Macro 80 Manual
49) Prime Computers	Subroutine Reference Guide

Appendix 1 - St.Venent torsion solution.

Appendix 1

The Use of PAFEC75 to solve the St.Venant torsion equation.

The routines listed in this appendix allow PAFEC75 level 3 to solve the St.Venant torsion equations by modifications made to the 2-dimensional heat conduction routines.

The program normally solves:

$$\boxed{k(\nabla^2 T) = -Q}$$

While St.Venants equation is:

$$\boxed{\begin{aligned} &\nabla^2\phi = -2G\dot{\theta} \\ &\frac{1}{G}(\nabla^2\phi) = -2\dot{\theta} \end{aligned}}$$

Thus the two problems are identical with the substitution of the values:

$$\boxed{\begin{aligned} &k \rightarrow 1/G \\ &Q \rightarrow -2\theta \\ &T \rightarrow \phi \end{aligned}}$$

To allow this, the value of k has to be redefined in the MATERIALS module as 1/G. The calculations assume G to be non-dimensional, so for a single material beam G must be set to unity. For beams made from more than one material the value of k should be set Gm/Gs, where Gm is the value of G for the main material making up the section and Gs is the value of G for the particular element. For example, a steel beam with an aluminium insert has a value of k for the steel section of 1 and for the aluminium section k=(81E9/26.5E9)=3.06.

The twist term $-2\dot{\theta}$ must be spread over the surface in the same way as a constant heat flux. $\dot{\theta}$ is assumed to be 1 which gives an equivalent heat flux of $2W/m^2$. This must be apportioned to each node with respect to the area of the elements connected to that node. This is achieved automatically within the modified routines produced by the author and requires no additional data.

The boundary conditions for St.Venants' equation are usually taken as Φ=0 on all external boundaries. This is equivalent to setting the temperature on all the boundary nodes to zero using the TEMPERATURES module. No heat flux can occur across lines of symmetry. Therefore, a symmetry line can be represented in the program by leaving the nodes unconnected.

The program automatically integrates Φ over the area of the cross section to give the torsion constant (section 6.3) and the result is output at the end of Phase 7. The integration technique is simple, the area of each triangular element is calculated and multiplied by the average value of Φ over the element.

For hollow sections, the interior must be filled with a hypothetical material with a low modulus. This is done by setting k=10000. Higher values may be used with discresion, but since the analysis is only single precision (32bit) and rounding errors can become very significant.

Comparison of results with previous workers.

Example 1. The Square section.

Analytical solution J=0.1406m (Timoshenko)

Results of other workers:

J= 0.1388 (Ely and Zienkiewicz : by relaxation)

= 0.1407 (Muskhilishvilli : series solution)

Appendix 1 - St.Venent torsion solution.

PAFEC Results:

J= 0.134 (128 elements)

= 0.1361 (1/4 model, equivalent to 400 elements)

= 0.1388 (1/4 model, equivalent to 512 elements)

= 0.1399 (1/4 model, equivalent to 2048 elements)

= 0.1377 (1/4 model, equiv. 400 elements, fine edge spacing 1 1 2 3 5)

= 0.1376 (1/4 model, equiv. 400 elements, fine edge spacing 1 2 3 5 8)

Maximum stresses: Timoshenko; 1.35xGO PAFEC75; 1.33xGO

Figure ap1.2 shows stress plots for this bar.

Example 2. The Equilateral triangle.

Analytical solution, J= 0.03849 (Timoshenko)

PAFEC75 Solution, J= 0.0376 (1/2 model, equiv. 200 elements)

Example 3. Ellipse.

Analytical solution: J= 0.3142 (Timoshenko)

PAFEC75 J= 0.3096 (1/4 model, equv. 800 elements)

Example 4. Bi-metalic square section bar.

Relaxation solution: J= 0.2358 (Ely and Zienkiewicz)

Series Solution: J= 0.2399 (Muskhelishvilli)

PAFEC75 J= 0.238

Example 5. Bi-metalic square section bar with hole.

Relaxation solution: J= 0.2138 (Ely and Zienkiewicz)

PAFEC75 J= 0.228

The PAFEC solution used a 'hole-material' with a value of G 1000 times lower than that of the structure.

Appendix 1 - St.Venent torsion solution.

Figure ap1.1 - Cross sections analysed.

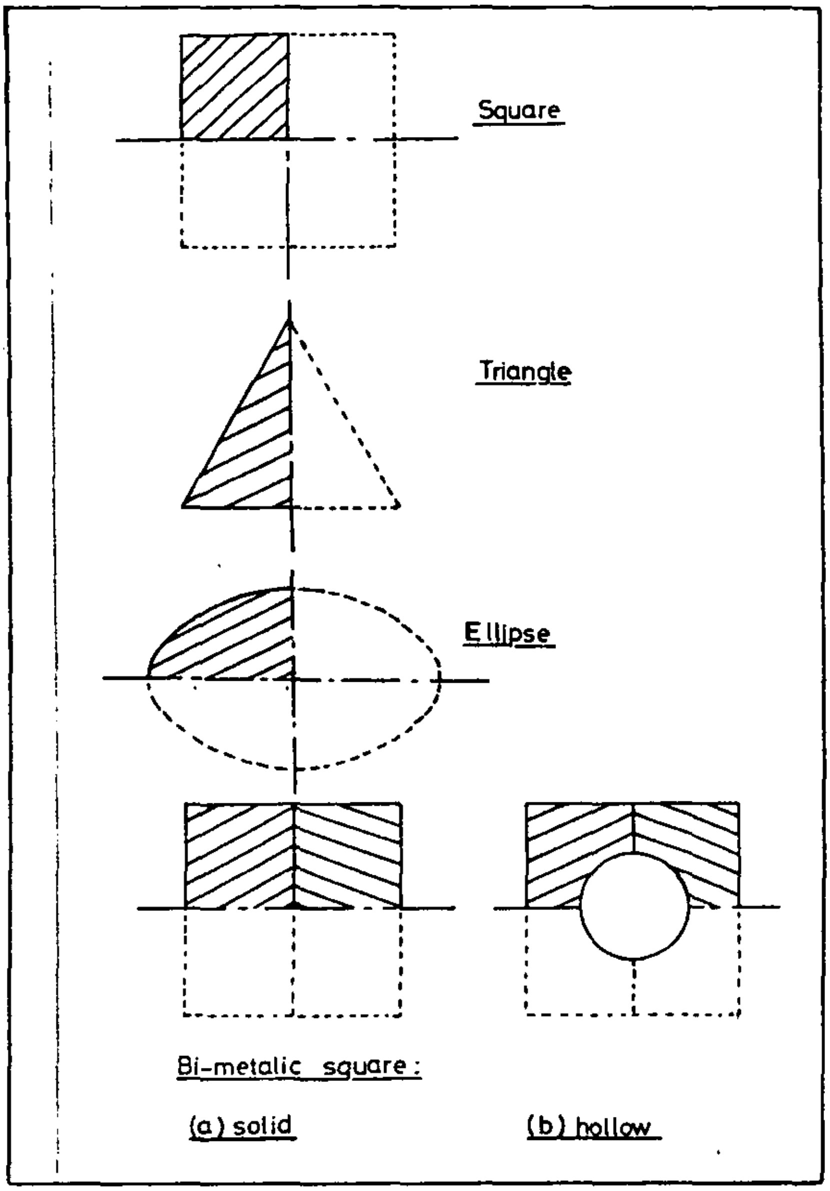
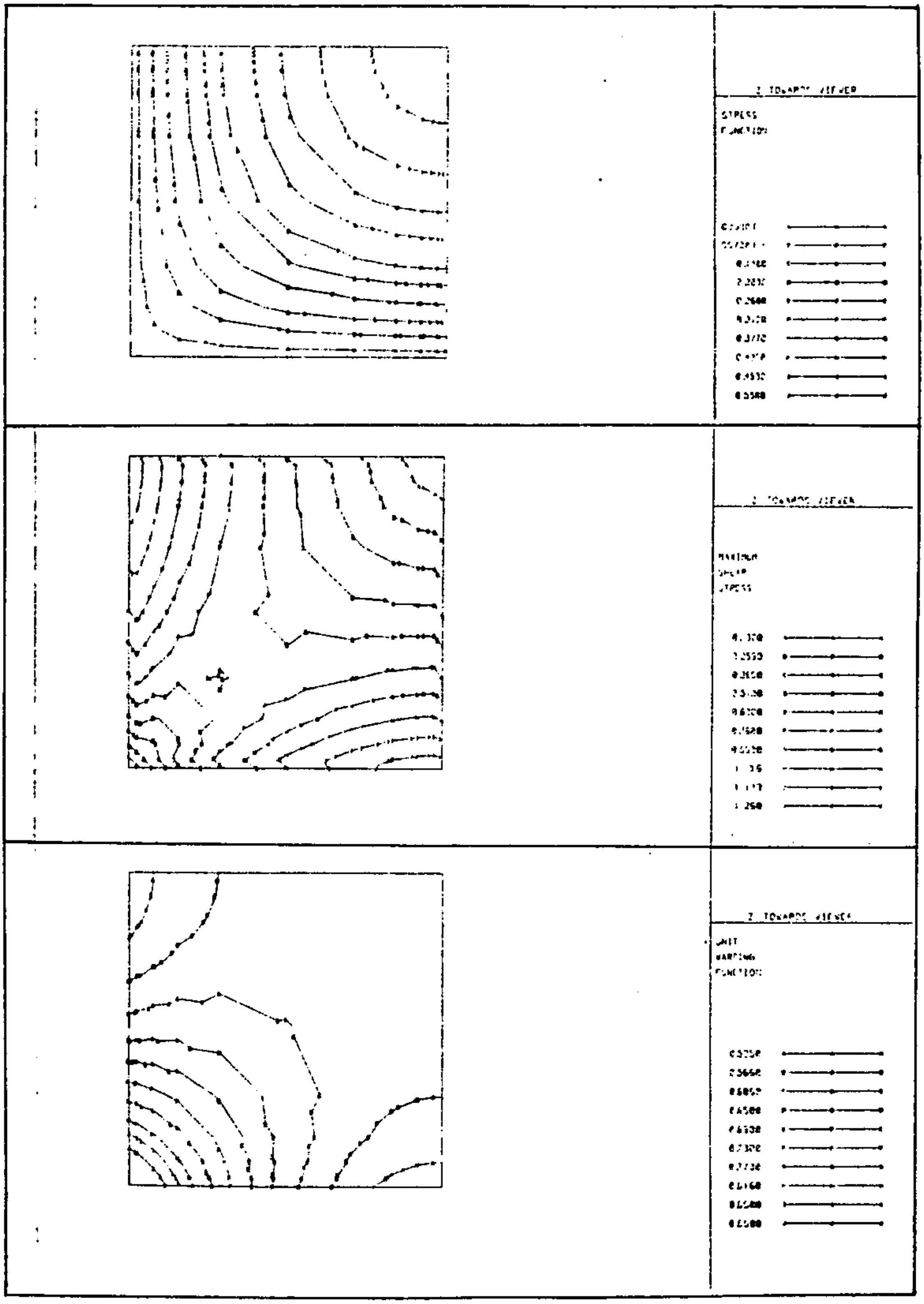


Figure ap1.2 - Pafec75 St.Venent Stress plots.



ST. VENANT TORSRON PROGRAM -
MODIFICATIONS TO PAFEC 75 ROUTINES

```
      SUBROUTINE R14120(CPDDC,PLO,WORK,IP)
      DIMENSION CPDDC(IP),PLO(IP),WORK(IP,7),L(3),X(3),Y(3)
      COMMON /IBASE/ IBASE(1000)
      COMMON BASE(33000)
C     CALLED BY R14100
C
C     INITIALISE KNOWN TEMPERATURES AND HEAT FLOWS FOR STEADY STATE CAL
C
C
C     OBTAIN TEMPERATURE MODULE (NO.53)
C
      ISTOP=0
      I32=IBASE(32)
      IF(I32.EQ.6) GOTO 100
      CALL R09800(53,1)
      CALL R09806(53,LMSP,JROWSP,ISPEC)
C
C
C     INITIALISE KNOWN PRESSURES FOR
C     LUBRICATION CALCULATION
C
C     OBTAIN PRESSURE MODULE (NO.36)
C
      GO TO 110
  100 CALL R09800(36,1)
      CALL R09806(36,LMSP,JROWSP,ISPEC)
C
C     BRING DOWN RESTRAINTS MODULE (41)--USED TO
C     SPECIFY PRESSURES OR TEMPERATURES ALONG A
C     LINE OR PLANE
C
  110 CALL R09800(41,1)
      CALL R09806(41,LM41,JR41,IPOS41)
      IF(LM41.EQ.0) GOTO 220
      ITOL=IBASE(28)
      IF(ITOL.LT.0) TOL=1/10.0**(-ITOL)
      IF(ITOL.GT.0) TOL=10.0**ITOL
      IF(ITOL.EQ.0) TOL=0
  102 FORMAT(1H ,4HTOL=,E10.3)
      CALL R09800(1,1)
      CALL R09806(1,LM1,JROW1,INODES)
      IEND=IPOS41+LM41-1
      IPOINT=IPOS41
  120 INODE=BASE(IPOINT)+.01
      IPLANE=BASE(IPOINT+1)+0.01
      IF(IPLANE.EQ.0) GOTO 220
      SPEC=BASE(IPOINT+4)
      IPOINT=IPOINT+5
      IF(INODE.EQ.0) GOTO 210
      IF(IPLANE.LE.6) GOTO 130
      WRITE(6,8) IPLANE
  130 CONTINUE
      IN=(INODE-1)*3+INODES-1
      GOTO (140,140,140,150,160,160),IPLANE
  140 ICONST=IPLANE
      GOTO 170
  150 ICONST=2
      GOTO 170
  160 ICONST=1
  170 ITWO=0
      IF(IPLANE.GT.3) ITWO=1
      IPRIME=IN+ICONST
      V2=BASE(IPRIME)
      IF(ITWO.EQ.0) GOTO 180
      IPRIME=IN+(9-ICONST-IPLANE)
      V3=BASE(IPRIME)
  180    DO 200 L1=1,IP
         IZ=(L1-1)*3+INODES-1
         IPRIME=IZ+ICONST
         V1=BASE(IPRIME)
         V1=ABS(V1-V2)
         IF(V1.GT.TOL) GOTO 200
         IF(ITWO.EQ.0) GOTO 190
         IPRIME=IZ+(9-ICONST-IPLANE)
         V4=BASE(IPRIME)
         V4=ABS(V3-V4)
         IF(V4.GT.TOL) GOTO 200
  190    CPDDC(L1)=0.0
         WORK(L1,1)=SPEC
         WORK(L1,2)=0.0
         WORK(L1,4)=SPEC
  200    CONTINUE
  210 IF(IPOINT.LT.IEND) GOTO 120
  220 IF(LMSP.EQ.0) GOTO 300
      IEND=ISPEC+LMSP-1
      IPOINT=ISPEC+1
  230 SPEC=BASE(IPOINT)
      ISTART=BASE(IPOINT+1)+.01
      IFIN=BASE(IPOINT+2)+.01
      ISTEP=BASE(IPOINT+3)+.01
      IPOINT=IPOINT+4
      IF(ISTART.EQ.0)GO TO 260
      IF(IFIN.GT.0)GO TO 240
C     GO THROUGH LOOP ONCE
      IFIN=ISTART
      ISTEP=1
  240 IF(IFIN.GT.IP)IFIN=IP
      IF(ISTEP.LE.0)ISTEP=1
      IFIN=ISTART+ISTEP*((IFIN-ISTART)/ISTEP)
         DO 250 L1=ISTART,IFIN,ISTEP
         CPDDC(L1)=0.0
         WORK(L1,4)=SPEC
         WORK(L1,2)=0.0
         WORK(L1,1)=SPEC
  250    CONTINUE
  260 CONTINUE
      NLIST=BASE(IPOINT)+0.01
      IPOINT=IPOINT+1
      IF(NLIST.EQ.0)GO TO 290
         DO 280 L1=1,NLIST
         INODE=BASE(IPOINT)+.01
         IPOINT=IPOINT+1
         IF(INODE.EQ.0)GO TO 280
         IF(INODE.GT.0.AND.INODE.LE.IP)GO TO 270
C          WRITE(6,6)INODE,IP
         GO TO 280
  270    CPDDC(INODE)=0.0
         WORK(INODE,4)=SPEC
         WORK(INODE,1)=SPEC
         WORK(INODE,2)=0.0
  280    CONTINUE
  290 CONTINUE
      IPOINT=IPOINT+1
      IF(IPOINT.LT.IEND)GO TO 230
  300 CALL R09800(53,5)
C ----- MODIFICATION FOR ST.VENANT TORSION     LJP 1/1/82  -------
C ----- GENERATE FLUX MODULE  54     ---------------------------------
--
C     GENERATE MODULE 130       --- ELEMENT AREAS
      IE=IBASE(39)
      IP=IBASE(3)
      CALL R09808(130,IE,1,IE,JROW,IPS130)
      CALL R09806(130,LM130,JRW130,IPS130)
      CALL R09800(1,1)
      CALL R09806(1,LM1,JROW1,IPOS1)
C     GENERATE MODULE 54
      CALL R09808(54,IP,2,2*IP,JROW,IPOS54)
      CALL R09806(54,LM54,JROW54,IPOS54)
C     CALL IN ELEMENTS MODULE 17
      CALL R09800(17,1)
      CALL R09806(17,LM17,JROW17,IPOS17)
C       WRITE(6,1100) LM130,JRW130,IPS130,LM54,JROW54,IPOS54
C      +,LM1,JROW1,IPOS1
C 1100  FORMAT(6I7)
      DO 1000 I=1,IE
      DO 1001 J=1,3
1001  L(J)=NYNT(BASE(IPOS17+4+J+(I-1)*8))
      DO 1002 J=1,3
      L2=IPOS1+(L(J)-1)*3
      L3=L2+1
      X(J)=BASE(L2)
1002  Y(J)=BASE(L3)
C       WRITE(6,1101) (L(I1),X(I1),Y(I1),I1=1,3)
C 1101  FORMAT(I7,2E15.3)
      AREA=ABS((X(2)*Y(3)-Y(2)*X(3)-X(1)*Y(3)+Y(1)*X(3)+X(1)*Y(2)-Y(1)*
X
     +(2))/2.)
      BASE(IPS130+I-1)=AREA
      DO 1003 J=1,3
      BASE(IPOS54+2*L(J)-2)=L(J)
1003  BASE(IPOS54+2*L(J)-1)=BASE(IPOS54+2*L(J)-1)+AREA/1.5
1000  CONTINUE
```

```
C ----- MAKE COPY OF TAGS MODULE ---------------------------------------
--
      CALL R09800(3,1)
      CALL R09806(3,LM3,JROW3,IPOS3)
      CALL R09808(129,LM3,1,LM3,JRW129,IPS129)
      DO 101 J=1,LM3
101   BASE(IPS129+J-1)=BASE(IPOS3+J-1)
C ---- MODIFICATION END ------------------------------------------------
-
      IF(LM54.EQ.0)GO TO 400
      CALL R09800(2,1)
      CALL R09806(2,LM2,JROW2,IPOS2)
      CALL NULL(PLO,IP,1)
      IF(IBASE(32).NE.6) GO TO 320
         DO 310 L1=1,IP
         IPRIME=IPOS54+L1-1
         IPRIM2=IPOS2+L1-1
         IDOF=BASE(IPRIM2)
         PLO(IDOF)=PLO(IDOF)+BASE(IPRIME)
  310    CONTINUE
      GOTO 370
C   320 WRITE(6,1)
320   CONTINUE
C       WRITE(6,2)
      JFIN54=IPOS54+LM54-1
      IPOINT=IPOS54
  330 NODE=BASE(IPOINT)+.01
      IPOINT=IPOINT+1
      IF(NODE.GT.0.AND.NODE.LE.IP)GO TO 340
      IF(NODE.EQ.0) GOTO 355
C       WRITE(6,4)NODE,IP
      ISTOP=1
      IPOINT=IPOINT+1
      GO TO 360
  340 IF(CPDDC(NODE).GT.0.5)GO TO 350
C       WRITE(6,5) NODE
  350 FLUX=BASE(IPOINT)
      IPOINT=IPOINT+1
      IPRIME=IPOS2+NODE-1
      IDOF=BASE(IPRIME)
      PLO(IDOF)=PLO(IDOF)+FLUX
C       WRITE(6,3)NODE,FLUX
      GOTO 360
  355 IPOINT=IPOINT+1
  360 IF(IPOINT.LT.JFIN54)GO TO 330
  370 CALL R09800(54,5)
  400 CONTINUE
      IF(ISTOP.EQ.1)STOP 10
C       WRITE(6,7)
      RETURN
    1 FORMAT(///,25X,
     +'2*THETADOT FLUX FOR ST VENANT TORSION CALCULATIONS')
    2 FORMAT(1H0,30X,18H  NODE        VALUE,/)
    3 FORMAT(1H ,30X,I5,E15.4)
    4 FORMAT(1H0,30X,12HNODE NUMBER ,I5
     +,30HIN FLUX MODULE IS ILLEGAL (IP=,I5,1H))
    5 FORMAT(1H0,32HWARNING FLUX INPUT GIVEN AT NODE ,I5,
     +34H WHICH HAS A SPECIFIED TEMPERATURE)
    6 FORMAT(1H0,13HWARNING NODE ,I5,
     +47H IN TEMPERATURE MODULE IS ILLEGAL ----- IGNORED)
    7 FORMAT(///)
    8 FORMAT(1H ,8HIPLANE =,I5,31HIS NOT VALID IN PRESSURE MODULE
     +,16H FOR LUBRICATION)
      END
      SUBROUTINE R61110
C-----COMMENT-----------------------------------------------------
C
C
C
C     ST VENANT TORSION STRESS FUNCTION    CALCULATION ROUTINE
C
C     THIS ROUTINE IS CALLED FROM R61130
C
C     R14100 INITIALISES WORKSPACE AREAS,CONTROLS THE MERGING
C
C     AND REDUCTION AND FINALLY THE BACK SUBSTITUTION TO OBTAIN
C
C     THE  SOLUTION
C
C
C
C-----COMMENT-END-------------------------------------------------
C
      DIMENSION L(3),PHIFD(3),COORD(3,3)
      COMMON/IBASE/IBASE(1000)
      COMMON BASE(33000)
      IP=IBASE(3)
      CALL R14100
      IF(IBASE(32).EQ.1) CALL R09800(5,5)
C-----BRING DOWN THE NON STRUCTURAL NODES MODULE
      CALL R09800(137,1)
C-----OBTAIN CO-ORDINATES MODULE 1
      CALL R09800(1,1)
      CALL R09806(1,LM,JROW,IPOS1)
C-----OBTAIN START ADRESSES OF PLO(M006) AND WORK(M076)
      CALL R09806(6,LM,JROW,IPOS6)
      CALL R09806(76,LM,JROW,IPOS76)
C-----OBTAIN DOF MODULE (M002)
      CALL R09800(2,2)
      CALL R09806(2,LM,JROW,IPOS2)
      IP=IBASE(3)
      CALL R09800(67,1)
      CALL R09806(67,LM67,JROW,IPOS67)
      IF(LM67.GT.0)GO TO 100
C-----COMMENT-----------------------------------------------------
C
C
C
```

```
C     CREATE MODULE 67-THIS STORES STEADY STATE TEMPERATURES FOR
C
C          (A)   PLOTTING
C
C          (B)   HEAT FLOW CALCULATION
C
C
C
C-----COMMENT-END-----------------------------------------------------
C
      CALL R09808(67,5*IP,1,5*IP,JROW,IPOS67)
  100 CO=-0.9E20
      WRITE(6,1)
      WRITE(6,2)
         DO 120 L1=1,IP
         IF(I09896(137,L1).EQ.0)GO TO 120
         IPRIME=IPOS76+L1-1
         TEMP=BASE(IPRIME)
         IF(TEMP.LT.CO)GO TO 110
         L2=IPOS1+(L1-1)*3
         L3=L2+2
C          WRITE(6,3)L1,(BASE(L4),L4=L2,L3),TEMP
         IPRIME=IPOS67+L1-1
         BASE(IPRIME)=TEMP
         GO TO 120
  110    IPRIME=IPOS2+L1-1
         IDOF=NYNT(BASE(IPRIME))
         L2=IPOS1+(L1-1)*3
         L3=L2+2
         IPRIME=IPOS6+IDOF-1
C          WRITE(6,4)L1,(BASE(L4),L4=L2,L3),BASE(IPRIME)
         JPRIME=IPOS67+L1-1
         BASE(JPRIME)=BASE(IPRIME)
  120    CONTINUE
      IF(IBASE(32).EQ.1)GO TO 140
C-----COMMENT---------------------------------------------------------
C
C
C
C     IF TRANSIENT CALCULATION TO FOLLOW COPY TEMPERATURE
C
C     FIELD INTO MODULE 6
C
C
C
C-----COMMENT-END-----------------------------------------------------
C
         DO 130 L1=1,IP
         IPRIME=IPOS6+L1-1
         JPRIME=IPOS67+L1-1
         BASE(IPRIME)=BASE(JPRIME)
  130    CONTINUE
C-----WRITE INITIAL TEMPERATURE FIELD TO FILE FOR PLOTTING
  140 IF(IBASE(34).NE.0)CALL R14135(BASE(IPOS67),IP)
```

```
C ---- MODIFIED ROUTINE FOR ST.VENANT TORSION   LJP 1/1/82 -----------
-----
C ----- OBTAIN ELEMENTS MODULE ------------------
      CALL R09800(17,2)
      CALL R09806(17,LM,JROW,IPOS17)
C ---- OBTAIN MATERIAL MODULE ----------------------------------
      CALL R09800(31,2)
      CALL R09806(31,LM,JROW,IPOS31)
C ----- OBTAIN PLATES.AND.SHELLS MODULE ------------------------------
----
      CALL R09800(27,2)
      CALL R09806(27,LM,JROW27,IPOS27)
C ---- OBTAIN AREAS MODULE 130 ---------------------------------------
-
      CALL R09800(130,1)
      CALL R09806(130,LM,JROW,IPS130)
C ---- OBTAIN MODULE 129  TAGS COPY  ---------------------------------
--
      CALL R09806(129,LM,JROW,IPS129)
      VOL=0.0
      IEL=0
      ITOPO=IPOS17+5
103   IEL=IEL+1
       IPROP=NYNT(BASE(ITOPO-2))
       I=0
 106   I=I+1
       IF(I.GT.10) CALL EXIT1
       IF(IPROP.NE.NYNT(BASE(IPOS27+JROW27*(I-1)))) GOTO 106
       MAT=NYNT(BASE(IPOS27+1+JROW27*(I-1)))
      RMATK=BASE(IPOS31+6+(MAT-1)*8)
      NON=ITOPO-1
      DO 105 I=1,3
105   L(I)=NYNT(BASE(ITOPO+I-1))
      PHI=0.0
      DO 104 J=1,3
      I=L(J)
      IPRIME=IPOS67+I-1
      L2=IPOS1+(I-1)*3
      L3=L2+1
      COORD(J,1)=BASE(L2)
      COORD(J,2)=BASE(L3)
      COORD(J,3)=1.0
      PHIFD(J)=BASE(IPRIME)
      IDOF=BASE(IPOS2+I-1)
      LPRIME=IPS129+IDOF-1
      IF(RMATK.GE.1000.0) BASE(LPRIME)=BASE(LPRIME)-0.001
104   PHI=PHI+BASE(IPRIME)
      PHIA=PHI/3.
C ------ CALCULATE SHEAR STRESSES FROM GRADIENTS ON ELEMENTS -------
      CALL MATDIV(DET,PHIFD,COORD,3,1)
      DO 108 I1=1,3
      I=L(I1)
      IF(RMATK.GE.1000.0) GOTO 108
      L3=IPOS67+IP+I-1
      L4=L3+IP
      BASE(L3)=BASE(L3)+PHIFD(2)
      BASE(L4)=BASE(L4)-PHIFD(1)
108   CONTINUE
C       IF(RMATK.GE.1000.0) GOTO 107
C ---- IF MATERIAL IS VERY WEAK I.E. A HOLE, DO NOT INTEGRATE ON THIS E
LEMENT
      AREA=BASE(IPS130+IEL-1)
      VOL=VOL+AREA*PHIA*2.
107   ITOPO=ITOPO+8
      IF(IBASE(39).GT.IEL) GOTO 103
      DO 109 I=1,IP
      IDOF=BASE(IPOS2+I-1)
      RNN=BASE(IDOF+IPS129-1)
      NNODE=((RNN-AINT(RNN))*1000.0)+0.5
      IF(NNODE.EQ.0) GOTO 109
      L1=IPOS67+I-1
      L2=L1+3*IP
      L3=L1+IP
      L4=L3+IP
      LW=L2+IP
      LX=IPOS1+(I-1)*3
      LY=LX+1
      BASE(L4)=BASE(L4)/NNODE
      BASE(L3)=BASE(L3)/NNODE
      BASE(L2)=SQRT(BASE(L3)**2+BASE(L4)**2)
      BASE(LW)=BASE(L1)+0.5*(BASE(LX)**2+BASE(LY)**2)
      ICOUNT=ICOUNT+1
      IF(ICOUNT.GE.50) WRITE(6,2)
      IF(ICOUNT.GE.50) ICOUNT=0
      WRITE(6,3)I,BASE(IPOS1+(I-1)*3),BASE(IPOS1+(I-1)*3+1),
     +(BASE(J),J=L1,LW,IP)
109   CONTINUE
      WRITE(6,1)
      WRITE(6,5) VOL
      CALL R09800(67,4)
      CALL R09800(137,5)
      IBASE(8)=5
C ---- TELL PHASE 8 TO DO 5 PLOTS: PHI,TAWX,TAWY,TAW,W
C ---- MODIFICATION END
      RETURN
    1 FORMAT(///)
2     FORMAT('1',120('-'),/,
     +'  Node',7X,'Global  Co-ordinates',7X,'Stress Function',7X,
     +'TawX',12X,'TawY',6X,'Maximum Absolute',4X,'Unit Warping',/,
     +' Number',12X,'X',10X,'Y',14X,'Value',41X,'Shear Stress',
     +7X,'Function',/,
     +' ',120('-'),///)
3     FORMAT(I6,E15.3,E12.3,5E17.4)
5     FORMAT(1H1,41('*'),/,' St. Venant Torsion Constant = ',E11.4,
     +/,' ',41('*'))
      END
      SUBROUTINE EXIT1
      WRITE(6,1)
```

```
1     FORMAT(1H ,'MATERIAL PROPERTY FOR ELEMENT ',I5,'GREATER THAN 10',
     +/,'PLATES.AND SHELLS MODULE IS PROBABLY MISSING')
      CALL EXIT
      END
      SUBROUTINE R19260 (POINTS,SCALE,IEL,STRMAX,STRMIN,MN,MNE,IPLOT,
     +                   ITYPE,RLIMIT,NCASE,TIME)
C-----COMMENT----------------------------------------------------------
C
C
C
C     MAIN CONTOUR PLOTTING ROURINE .
C
C     CALLS R19261 TO DETERMINE RANGE OF VALUES TO BE PLOTTED,
C
C     PRINTS CONTOUR KEY HEADING ,
C
C     TEN CONTOURS ARE USUALLY DRAWN AT ROUGHLY EQUAL INTERVALS, THE
C
C     ACTUAL VALUE PLOTTED BEING TWO SIGNIFICANT DIGITS FOLLOWED BY
C
C     ZEROS .
C
C     FOR EACH CONTOUR ALL ELEMENTS ARE LOOPED THROUGH. R19262 IS
C
C     CALLED TO SEE IF THE CONTOUR PASSES THROUGH THE CURRENT ELEMENT.
C
C     IF IT DOES R19263 I
C     AND DRAW THE CONTOUR.
C
C     FINALLY THE CONTOUR VALUE
C
C
C-----COMMENT-END------------------------------------------------------
C
      DIMENSION   POINTS(3,1), SCALE(8),    IEL(100),  NODES(5,5)
      COMMON/IBASE/IBASE(1000)
      COMMON/C19260/ IE,XP,YP,NCONT,ILINE,IPLOT1,IMID,RNEGL,SMAX,
     + IPOWER,RANGE,DIFF,ASTRSS,XK,YK,IHEAD,NDIG,NLDIG,JCNT,NGROUP,
     + LM17,JROW17,IPOS17,LM18,JROW18,IPOS18,NOCHAR,L2,IPRIME,IPOINT,
     + JPOINT,KPOINT,INE,NE, IERN,IT,YSPACE,POWER,STRSS,STRSB
      COMMON BASE(100)
      REAL*8 RANGE1
      IE = IBASE(9)
      XP = SCALE(5)
      YP = SCALE(6)
C     POSSIBLE RANGE IS 1 TO 50 BUT
C     IF GREATER THAN 25 KEY MAY BECOME SQUASHED.
C     IPLOT IS DECODED - 1ST DIGIT ONLY REQUIRED FOR CONTOUR
C     PLOTTING - POINTS TO LOCATION IN ROW OF MODULES MN
C     OF VALUE TO BE PLOTTED
C     RLIMIT TO BE RENAMED AS PLTSCL - USER DEFINED
C     VALUE OF MAX CONTOUR
      NCONT = 10
      WRITE (6,1) ITYPE,NCONT
      ILINE = NCONT-4
      IPLOT1 = IPLOT/10
      IMID = -1
      IF (RLIMIT.EQ.0.0) GOTO 100
      RNEGL = -RLIMIT
      WRITE (6,2 ) RLIMIT
  100 CALL R19261 (POINTS,MN,IPLOT1,ITYPE,STRMIN,STRMAX,RLIMIT,NCASE)
      IF (ABS(STRMIN-STRMAX).GT.1.0E-10) GOTO 110
      WRITE (6,3 )
      RETURN
C-----COMMENT----------------------------------------------------------
------
C
C     THE FOLLOWING DETERMINES IPOWER, THE POMULTIPLES OF + OR - 6.
C
C-----COMMENT-END------------------------------------------------------
--------
  110 SMAX = STRMAX
      IF (ABS(STRMIN).GT.ABS(STRMAX)) SMAX=STRMIN
      PRIME = ALOG10(ABS(SMAX))
      IPOWER = PRIME
      IPOWER = IPOWER/5
      IPOWER = IPOWER*6
      RANGE1= ABS(STRMAX-STRMIN)
      DIFF = RANGE1/(NCONT-1)
      POWER = IPOWER
      IF(IAB(IPOWER).GE.6) RANGE1 = RANGE1/10.**POWER
      ASTRSS = STRMIN
      XK = XP-6.2
      YK = YP-7.1
      CALL POINT(XP,4.2)
      CALL JOIN(XP-6.5,4.2)
      CALL PLOTCL(XP-6.0,3.7,11HLOAD CASE =,11)
      CALL PLOTNI(XP-2.5,3.7,NCASE)
      IHEAD = ITYPE-30+NCASE
C    LINE BELOW MODIFIED PLOT TYPE 36 WAS GOTO 170,IS NOW GOTO 140
C    ALSO 141 WAS 190, 142 WAS 200, 143 WAS 210
      GOTO (120,130,135,150,150,160,140,141,142,143,144,240
     +,220,220,220,230),IHEAD
  120 CALL PLOTCL (XK,YK-4.5,7HMAXIMUM     , 7)
      CALL PLOTCL (XK,YK-5.0,9HPRINCIPAL   , 9)
      CALL PLOTCL (XK,YK-5.5,5H(MOST       , 5)
      CALL PLOTCL (XK,YK-6.0,9HPOSITIVE)   , 9)
      CALL PLOTCL (XK,YK-6.5,6HSTRESS      , 6)
      GOTO 240
  130 CALL PLOTCL (XK,YK-4.5,7HMINIMUM      , 7)
      CALL PLOTCL (XK,YK-5.0,9HPRINCIPAL    , 9)
      CALL PLOTCL (XK,YK-5.5,5H(MOST        , 5)
      CALL PLOTCL (XK,YK-6.0,9HNEGATIVE)    , 9)
      CALL PLOTCL (XK,YK-6.5,6HSTRESS       , 6)
      GOTO 240
  135 CALL PLOTCL(XK,YK-4.5,7HLARGEST,7)
      CALL PLOTCL(XK,YK-5.0,8HABSOLUTE,8)
      CALL PLOTCL(XK,YK-5.5,9HPRINCIPAL,9)
      CALL PLOTCL(XK,YK-6.0,6HSTRESS,6)
      GOTO 240
  143 CALL PLOTCL (XK,YK-5.5,7HMAXIMUM      , 7)
      CALL PLOTCL (XK,YK-6.0,5HSHEAR        , 5)
      CALL PLOTCL (XK,YK-6.5,6HSTRESS       , 6)
      ITYPE=36
      GOTO 240
140   CALL PLOTCL(XK,YK-4.5,6HSTRESS,6)
      CALL PLOTCL(XK,YK-5.0,8HFUNCTION,8)
      ITYPE=36
      GOTO 240
141   CALL PLOTCL(XK,YK-4.5,6HTAW XZ,6)
      CALL PLOTCL(XK,YK-5.0,5HSHEAR,5)
      CALL PLOTCL(XK,YK-5.5,6HSTRESS,6)
      ITYPE=36
      GOTO 240
142   CALL PLOTCL(XK,YK-4.5,6HTAW YZ,6)
      CALL PLOTCL(XK,YK-5.0,5HSHEAR,5)
      CALL PLOTCL(XK,YK-5.5,6HSTRESS,6)
      ITYPE=36
      GOTO 240
144   CALL PLOTCL(XK,YK-4.5,4HUNIT,4)
      CALL PLOTCL(XK,YK-5.0,7HWARPING,7)
      CALL PLOTCL(XK,YK-5.5,8HFUNCTION,8)
      ITYPE=36
      GOTO 240
  150 CALL PLOTCL (XK,YK-6.0,4HHOOP         , 4)
      CALL PLOTCL (XK,YK-6.5,6HSTRESS       , 6)
      GOTO 240
  160 CALL PLOTCL (XK,YK-5.5,3HVON          , 3)
      CALL PLOTCL (XK,YK-6.0,5HMISES        , 5)
      CALL PLOTCL (XK,YK-6.5,9HCRITERION    , 9)
      GOTO 240
  170 CALL PLOTCL (XK,YK-6.5,11HTEMPERATURE ,11)
      GOTO 240
  180 GOTO 240
C     HEADING FOR STREAMLINES IN POTENTIAL FLOW
  190 CALL PLOTCL (XK,YK-5.0,9HTRANSIENT    , 9)
      CALL PLOTCL (XK,YK-5.5,11HTEMPERATURE,11)
      CALL PLOTCL (XK,YK-6.0,7HAT TIME,7)
      CALL PLOTNO (XK+1.5,YK-6.5,TIME,0.3,4)
  200 GOTO 240
C     HEADING FOR EQUIV STRAIN IN PLASTICITY PROBLEMS
  210 GOTO 240
C     HEADING FOR EQUIV CREEP STRAIN IN PLASTICITY PROBLEMS
  220 GOTO 240
C     HEADING FOR YIELD LIN FOR EACH INCREMENT
  230 CALL PLOTCL(XK,YK-6.5,8HPRESSURE,8)
  240 DO 340 L1 = 1,NCONT
      IF (ITYPE.EQ.41) GOTO 250
      IF (L1.GT.1) ASTRSS = ASTRSS+DIFF
  250 GOTO 260
C     STE CONTOUR VALUE TO YIELD STRAIN, MOVE PONTER TO NEXT LOAD
C     INCREMENT BLOCK IN YIELD STRSAIN MODULE
  260 ISIGN = 1
C
C     COMPUTE STRSS - THE ACTUAL STRESS TO BE PLOTTED.
C     THIS IS A PRNUMBER WITH
```

```
C     (5-NDIG) SIGNIFICANT FIGURES
      NDIG = 3
      RANGE1 = ABS(RANGE1/STRMAX)
      IF(RANGE1.LT.15.) NDIG = 2
      IF (ABS(ASTRSS).LT.1.0E-12) ASTRSS = 1.0E-12
      IF (ASTRSS.LT.0.0) ISIGN = -1
      NLDIG = INT(ALOG10(ABS(ASTRSS)))
      ISTRSS = ASTRSS*0.1**(NLDIG-NDIG)
      STRSS = (ISTRSS*10.**(NLDIG-NDIG))
      JCNT = 0
  270 JCNT = JCNT+1
      NGROUP = IEL(JCNT)
      IF (NGROUP.EQ.0.AND.JCNT.NE.1) GOTO 310
      CALL R09806 (17,LM17,JROW17,IPOS17)
      CALL R09806 (18,LM18,JROW18,IPOS18)
C--------COMMENT-------------------------------------------------
   C
C
C
C     EACH CONTOUR LINE IS DRAWN SOLID WITH A DIFFERENT LETTER OF THE
C     ALPHABET.
C     WITH NOCHAR SET TO 11 THE FIRST LETTER IN THE N.P.L. SYSTEM
C     IS LETTER 'A'.
C
C
C-----COMMENT-END--------------------------------------------------
C
      CALL CRSIZE(0.2)
      NOCHAR = L1+10
C-----COMMENT-----------------------------------------------------
C
C
C
C     LOOP THROUGH ALL ELEMENTS  (N.B. IF IPOINT = 0 ELEMENT
C
C     NUMBER L2 DOES NOT EXIST) .
C
C     CHECK IERN , DO NOT PLOT IF
C          IT = 30 (30100 , 30200 - LUMPED MASS AND SPRING ELEMENTS )
C
C          IT = 34 ( 34000 SERIES - BEAM ELEMENTS )
C
C          IT = 37 ( 3-D ELEMENTS )
C
C          42 (42000 SERIES )
C
C          OR BOUNDARY LAYER ELEMENT 39310 , 39410 , 39510
C
C     IF ELEMENT IS IN CURRENT GROUP CALL R19262 TO DECIDE
C
C     IF CURRENT CONTOUR PASSES THROUGH THIS ELEMENT .
C
C     IF IT DOES R19263 IS CALLED TO SET UP THE ARRAY NODES
C
```

```
C     BY CALLING R35000 .
C
C     R19264 THEN LOCATES AND DRAWS THE CORRECT CONTOUR THROUGH
C
C     THE ELEMENT
C
C
C-----COMMENT-END---------------------------------------------------
C-
      DO 300 L2 = 1,IE
      IPRIME = IPOS18+L2-1
      IPOINT = BASE(IPRIME)+0.1
      IF (IPOINT.EQ.0) GOTO 300
      JPOINT = IPOS17+IPOINT-1
      KPOINT = JPOINT-JROW17-1
      INE = BASE(KPOINT) + 0.1
      NE     = BASE(JPOINT)+0.1
      IERN = BASE(JPOINT+2)+0.1
      IT = IERN/1000
      IF (IT.EQ.30.OR.IT.EQ.34.OR.IT.EQ.42.OR.IT.EQ.37)
     +GOTO 300
      IF (IERN.EQ.39310.OR.IERN.EQ.39410.OR.IERN.EQ
     +.39510) GOTO 300
      IF (IERN.EQ.36120) IERN = 36100
      IF (IERN.EQ.36100) INE = 3
      IF (IERN.NE.36220.AND.IERN.NE.36230.AND.IERN.NE.36240.AND.IERN.NE

     + 36250) GO TO 280
      IERN = 36200
      INE = 4
  280 ISZ = INE/4.4 + 2
      IF (NGROUP.EQ.0) GOTO 290
      MGROUP = BASE(JPOINT+1)+0.1
      IF (MGROUP.NE.NGROUP) GOTO 300
  290 CALL R19262 (MN,IPLOT1,ITYPE,NE,STRSS,IRET,NCASE,INE)
      IF (IRET.NE.1) GOTO 300
      IPRIME = JPOINT-JROW17-1
      CALL R19263 (NODES,ISZ,INE,IMID,IERN)
      CALL R19264 (POINTS,MN,MNE,NODES,ISZ,IPLOT1,ITYPE,NE,STRSS,NCASE)
       CALL TYPENC (NOCHAR)
  300 CONTINUE
      GOTO 270
  310 IF (L1.GT.1) GOTO 320
      YSPACE=0.6
      IF(NCONT.GT.15) YSPACE=0.7
      IF(NCONT.GT.22) YSPACE=0.5
      IF(NCONT.GT.34) YSPACE=0.35
  320 CALL CRSIZE(0.3)
      IF (IAB(IPOWER).LT.6) GOTO 330
      POWER = IPOWER
      STRSS = STRSS/10.**POWER
      NDP = 1
      STRSB = STRSS
      IF(ABS(STRSS).LT.1.0E-12) STRSB = 1.0E-12
      IPRIME = ALOG10(ABS(STRSB))
      IF(STRSB.LT.10.) NDP = -IPRIME+4-NDIG
      XK = XP-5.1-NDP*0.2
      IF (L1.NE.1) GOTO 330
      CALL PLOTCL(XP-6.2,YK-7.0,8HMULTIPLY,8)
      CALL PLOTCL(XP-6.2,YK-7.5,5HBY 10,5)
      CALL CRSIZE(0.2)
      CALL PLOTNI(XP-5.1,YK-7.3,IPOWER)
  330 YL=YK-7.5-L1*YSPACE
      CALL PLOTNO (XP-4.5,YL,STRSS,0.3,4)
      CALL CRSIZE(0.2)
      CALL POINT(XP-4.0,YL)
      CALL TYPENC(NOCHAR)
      CALL POINT (XP-4.0,YL)
      CALL JOIN(XP-2.5,YL)
       CALL TYPENC(NOCHAR)
      CALL POINT (XP-2.5,YL)
      CALL JOIN(XP-1.0,YL)
       CALL TYPENC(NOCHAR)
  340 CONTINUE
      RETURN
    1 FORMAT(///,10X,47HCONTOUR PLOTTING ROUTINE ENTERED FOR PLOT TYPE
,
     +I3,/,10X,I3,28H CONTOURS ARE TO BE PLOTTED,         )
    2 FORMAT (10X,44HREQUESTED PLOTTING RANGE IS UPTO MAXIMUM OF ,E12.3
)
    3 FORMAT(/,20X,38HABOVE PLOTTING RANGE UNSATISFACTORY,      ,
     +/,42H  ******  THIS PLOT ABANDONED  ******              )
      END
```

Finite element mesh digitising program.

This program was designed to aid in the rapid development of simple finite element models. The program works on the Prime 400 computer using a sigma S5660 colour plotting terminal and a bitpad digitising tablet. The program is written entirely in FORTRAN 66 and utilises GINO plotting routines. A few Primos subroutines are also used therefore precluding the possibility of transfer to a different machine.

The program is simple and interactive, but presently allows no editing of data. The program is supported at all stages by HELP messages and should never leave the user stranded. All input statements check for errors and therefore stop the program 'crashing' when faulty data is input.

The program works at two command levels. The first is a 'Menu' (list) of sub commands which allows any of the sub-menu's to be accessed.

On initialisation of the program the computer responds with the request COMMAND:. Typing Help will list out the main menu of commands with an explanation (Figure ap2.1 lists all the menu's).

If an initial command other than Axis or Resume is typed then an error will be flagged since nothing meaningful can be achieved before the bitpad dimensions are set or a list of nodal coordinates is available from a previous session.

Reading data from the bitpad.

The bitpad sends a continuous stream of coordinate data while the pen or cursor is within 10mm of the digitising surface. This is sent in the format x,y,{0 or 1}, where x and y are sent as integer numbers in units of 0.1mm and the final digit is one if the pen is in contact and zero otherwise.

The bitpad information is read by the subrotine BPOINT into a 3 element array A. If A(3)=0 the coordinate is ignored and a second one is read until A(3)=1. Since the string of coordinates continues to arrive after the required one has been read it is necessary to make the computer ignore all the exess data. This is simply done by ignoring any coordinate which is within 5mm of the previous one.

Defining the coordinates and scaling of the digitised data using the Axes command.

The Axes command requests the definition of some coordinates and then their location on the bitpad. There are different definitions for the 2D and 3D programs (Figures ap2.2 and ap2.3)

2D definition (ap2.2). The computer requests the true coordinates of the points at the bottom left and top right of the drawing. Then it asks the user to locate the points on the bitpad, i.e. press down the pen on the equivalent points on the drawing.

3D definition (ap2.3). The computer requests the true x,y,z coordinates of the six points shown (in the correct order) and then their location on the bitpad.

Generating nodes.

With the axis system defined it is now possible to generate nodes, by first issuing the command Nodes. The computer will respond with a request to switch input to the bitpad. This done it is a simple matter of locating the required node on the drawing and pressing down the pen. The computer then reads the bitpad coordinate and calculates the true coordinate. Note that for the 3D program the node must be located on both drawings, the left hand one first.

Three dedicated areas are set asside on the bit pad (and nodes cannot be generated at these points). Their purpose is to allow control of the program.

They are:

1) Accept node.
2) Reject last node.
3) Exit node generation - return to main menu.

Generation of Elements and Pafblocks.

When node generation is complete, and the program has returned to the main menu, the commands for Elements or Pafblocks can be issued. The two routines are very similar except for a request for the number of elements along each side of the Pafblocks.

The computer first requests the type of element required (help will give a list), the computer knows the number of sides each element has. If pafblocks are being used the computer will request the type of pafblock to be used (only 1 and 2 are supported - Help will give advice on the choice).

A cursor will now be put up on the graphics screen, this is moved using the cursor control pad. The cursor is used to locate the nodes to define an element or pafblock. When the cursor is positioned the node is selected by pushing the <spacebar> key. The node selected will be the one closest to the cursor position. Note that on this menu no <cr> is required after a key is pressed.

If the wrong node is chosen it may be rejected immediately after selection by pressing the R key. There is presently no way of correcting topology errors later on. If an element has been defined wrongly, and this is noticed before the last node is picked, the whole element can be rejected by pressing B. To select a different element type press E and to set the property number press P. When all elements or pafblocks are complete type F to return to the main menu.

The basic data for the model is now complete. To create a PAFEC data file type Stop. The computer will respond with a request for a file name and the data will be written to this file. For a run to be made additional modules will have to be edited in by hand. e.g. Plates.and.shells, Beams, Loads, Restraints.

To pause in the middle of a model generation use the command Quit, a dump file will then be generated which can be Resumed at a later date.

Figure ap2.4 shows the computer terminal layout.

Appendix 2 - Mesh generator.

Figure ap 2.1 - Menu's

```
              MENU of Commands at this level is as follows :
        N     -     Generate node information
        A     -     Set bit-pad drawing dimensions (AXES)
        E     -     Generate element information
        P     -     Generate PAFBLOCKS information
        D     -     Draw all accepted information on screen
        S     -     Stop generating session, output data
        Q     -     QUIT and dump all data generated to a file
        R     -     RESUME session using data from previous dump
        T     -     Change teminal type to tektronix

              Menu of commands at ELEMENT level
        <spacebar>  Select node at cursor
        R     -     Reject last selected node
        E     -     Redefine element type
        F     -     Finish element generation, Return to Main Menu
        P     -     Set element property number
        B     -     Begin definition of present element again

            Element types are as follows :
     44200    -     4 noded bending and in-plane plate
     44210    -     8 noded    "       "       "
     44100    -     3 noded    "       "       "
     44110    -     6 noded    "       "       "
     41320    -     3 noded hybrid stress plate bending
     34000    -     2 noded simple beam element
     34200    -     2 active+2 offset defining nodes, offset beam

     43210    -     8 Noded curved shell semi-loof element
     43310    -     3 Noded curved beam semi-loof element
     43110    -     6 Noded curved shell semi-loof element
     45210    -     8 Noded thick shell element
     45110    -     6 Noded thick shell element
     39***    -     Series of elements for heat transfer calcs
     36***    -     Series of isoparametric shell elements
```

Figure ap2.2 - 2D Axes definition.

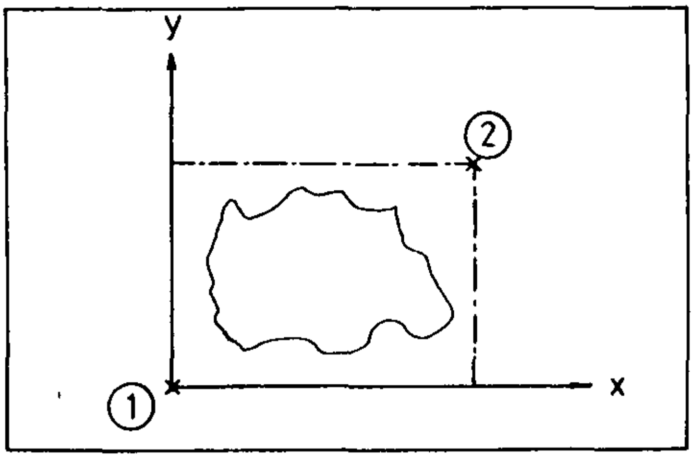


Figure ap2.3 - 3D Axes definition.

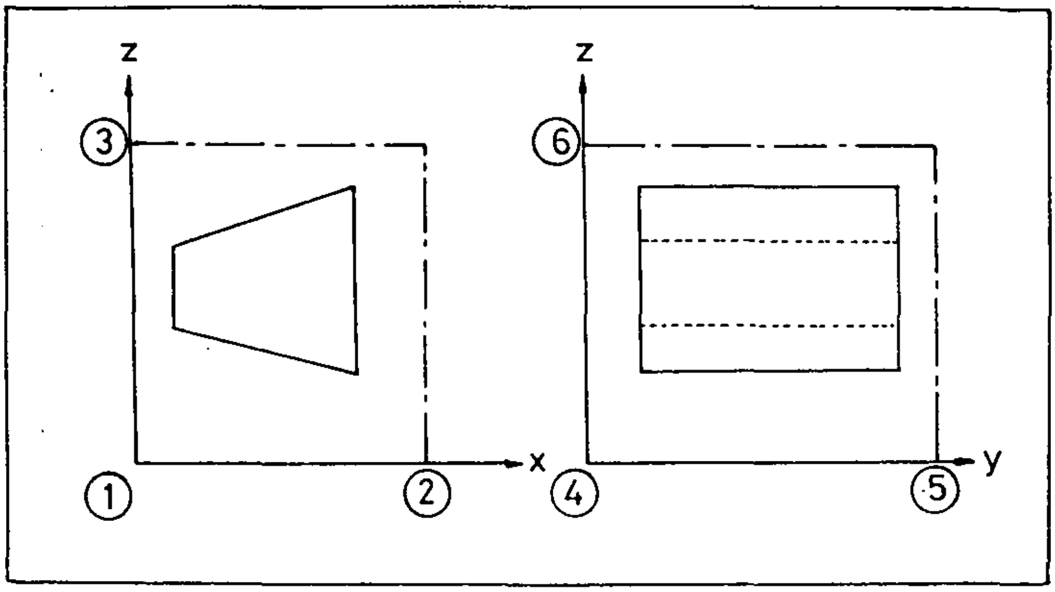
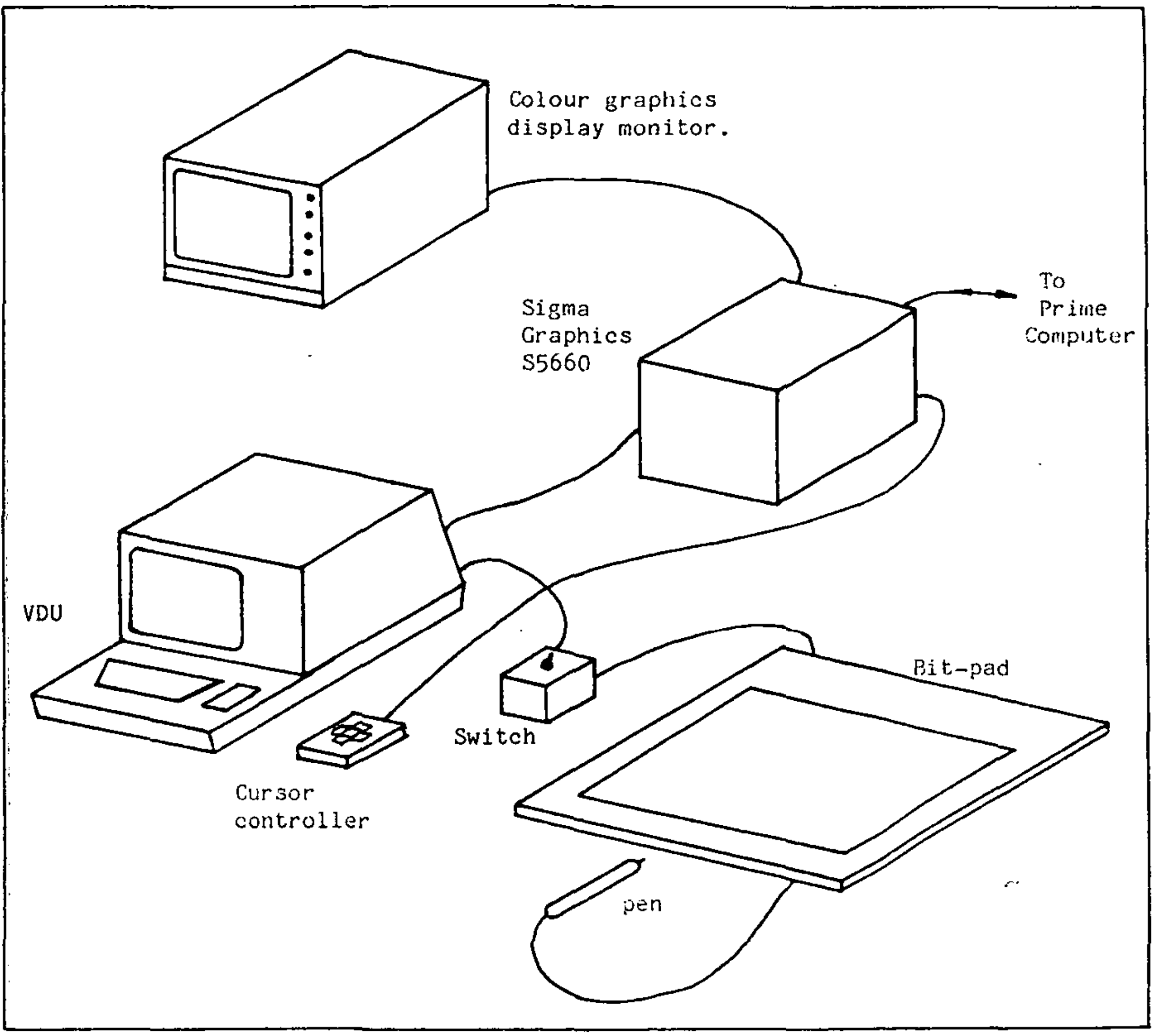


Figure ap2.4 - Terminal layout.



COMMON BLOCK DETAILS FOR PAFMESH3 PROGRAM
(FOR $INSERT TT)LJP)PAFCOM3)

```
 INTEGER*2 A(3),B(3),C(3),NN,EN,DUPLX$,NEN,TOLE,OFIL(8),PN
 REAL*4 CPX(6),CPY(6),CPZ(6),PSCAL,CXSCAL,CYSCAL,CZSCAL,VP(3)
 LOGICAL LOG,CLOG,RLOG,LAXIS,ELOG,REFLOG(100),ERLOG
 INTEGER*4 TOPO(11,100),IEL,TOPOP(14,100),AXIS(12)
 COMMON/BLK1/ A,B,C,NN,TOLE,PSCAL,CXSCAL,CYSCAL,CZSCAL,CPX,CPY,CPZ
+,LOG,CLOG,RLOG,LAXIS,X(100),Y(100),Z(100),TOPO,ELOG,AXIS
+,EN,IEN,NEN,OFIL,XP(100),YP(100),LINE(80),IEL,TOPOP,PN,NPN
+,REFLOG,TANTHE,THETA,ALPHA,BETA,VP,AXES(12),XPMIN,YPMIN,PCZ
```

PAFMESH3 PROGRAM

```
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      DATA VP(1),VP(2),VP(3)/1.0,1.0,1.0/
      CALL S5660
      CALL WINDOW(2)
      CALL CHASIZ(3.,2.)
      TOLE=5
      EN=1
      NN=1
      PN=1
1     CALL TNOUA('Command : ',10)
      READ(1,100)IA
100   FORMAT(A1)
      LOG=.FALSE.
      IF(IA.EQ.'N') CALL NODES
      IF(IA.EQ.'E') CALL ELEM
      IF(IA.EQ.'P') CALL PAFBL
      IF(IA.EQ.'A') CALL SAXIS
      IF(IA.EQ.'D') CALL DRAW
      IF(IA.EQ.'Q') CALL EXIT2
      IF(IA.EQ.'R') CALL SAVE
      IF(IA.EQ.'T') CALL DEVEND
      IF(IA.EQ.'T') CALL T4010
      IF(IA.EQ.'T') CALL WINDOW(2)
      IF(IA.EQ.'T') LOG=.TRUE.
      IF(IA.EQ.'V') CALL DEVEND
      IF(IA.EQ.'V') CALL TREND
      IF(IA.EQ.'V') CALL WINDOW(2)
      IF(IA.EQ.'V') LOG=.TRUE.
      IF(IA.EQ.'V') CALL TNOU('VDU ONLY SUITABLE FOR NODE PLOTTING',35)
      IF(IA.EQ.'H') CALL HELP1
      IF(IA.EQ.'S') CALL EXIT1
      IF(.NOT.LOG) CALL TNOU('Command not known',17)
      GOTO 1
      END
      SUBROUTINE SAXIS
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      LOG=.TRUE.
      CALL TNOU('Enter point coordinates as shown on instructions',48)
      DO 1 IP=1,6
      ENCODE(4,110,OFIL) IP
110   FORMAT(I4)
      CALL TNOUA('Point ',6)
      CALL TNOUA(OFIL,4)
      CALL TNOUA(' :  ',4)
1     READ(1,*) CPX(IP),CPY(IP),CPZ(IP)
      CALL TNOU('Switch to bit-pad',17)
      I=DUPLX$(-1)
      CALL DUPLX$(OR(:140000,I))
      DO 4 IP=1,6
      CALL TNOUA('Locate point ',13)
      WRITE(1,110) IP
      CALL BPOINT
      IP1=((IP-1)*2)+1
      IP2=IP1+1
      AXIS(IP1)=A(1)
4     AXIS(IP2)=A(2)
      TANTH1=((AXIS(4)-AXIS(2))/(AXIS(3)-AXIS(1)))
      TANTH2=((AXIS(1)-AXIS(5))/(AXIS(6)-AXIS(2)))
      TANTH3=((AXIS(10)-AXIS(8))/(AXIS(9)-AXIS(7)))
      TANTH4=((AXIS(7)-AXIS(11))/(AXIS(12)-AXIS(8)))
      TANTHE=(TANTH1+TANTH2+TANTH3+TANTH4)/4.
      THETA=ATAN(TANTHE)
      DO 999 K=1,12
999   AXES(K)=AXIS(K)
      CXSCAL=(CPX(2)-CPX(1))/(SQRT((AXES(3)-AXES(1))**2+(AXES(4)-AXES(2
)
     +)**2))
      CYSCAL=(CPY(5)-CPY(4))/(SQRT((AXES(9)-AXES(7))**2+(AXES(10)-AXES(
8
     +))**2))
      CZSCL1=(CPZ(3)-CPZ(1))/(SQRT((AXES(6)-AXES(2))**2+(AXES(2)-AXES(5
     +))**2))
      CZSCL2=(CPZ(6)-CPZ(4))/(SQRT((AXES(12)-AXES(8))**2+(AXES(11)-AXES
     +(7))**2))
      CZSCAL=(CZSCL1+CZSCL2)/2.
      PCX=(CPX(1)+CPX(2))/2.
      PCY=(CPY(4)+CPY(5))/2.
      PCZ=(CPZ(1)+CPZ(2)+CPZ(4)+CPZ(6))/4.
      CALL TNOU('Switch to terminal',18)
      CALL DUPLX$(I)
      LAXIS=.TRUE.
      CALL PLSET
      RETURN
      END
      SUBROUTINE BPOINT
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
1     READ(1,*) A
      IF(A(3).NE.1) GOTO 1
      IF(IABS(A(1)-B(1)).LT.TOLE.AND.IABS(A(2)-B(2)).LT.TOLE)GOTO 1
      B(1)=A(1)
      B(2)=A(2)
      CALL TNOUA('',1)
      RETURN
      END
      SUBROUTINE NODES
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
```

```
      IF(.NOT.LAXIS) CALL TNOU('Axes not set',12)
      LOG=.TRUE.
      IF(.NOT.LAXIS) RETURN
      CALL TNOU('Switch to bit-pad',17)
      I=DUPLX$(-1)
      CALL DUPLX$(OR(:140000,I))
1     CALL BPOINT
      CALL BITCOM
      IF(CLOG) GOTO 1
      NN=NN-1
      IF(RLOG) GOTO 2
      C(1)=A(1)
      C(2)=A(2)
      CALL BPOINT
      CALL BITCOM
      X(NN)=((C(1)-AXIS(1))*COS(THETA)+(C(2)-AXIS(2))*SIN(THETA))*CXSCA
L
     ++CPX(1)
      Y(NN)=((A(1)-AXIS(7))*COS(THETA)+(A(2)-AXIS(8))*SIN(THETA))*CYSCA
L
     ++CPY(4)
      Z(NN)=(CPZ(1)+CPZ(4))/2.+CZSCAL*((C(2)-AXIS(2))-(C(1)-AXIS(1))*TA
N
     +THE+(A(2)-AXIS(8))-(A(1)-AXIS(7))*TANTHE)*COS(THETA)/2.
      CALL PLOTCO
      CALL PPOINT
      IF(RLOG) GOTO 2
      GOTO 1
2     CALL TNOU('Switch to terminal',18)
      CALL DUPLX$(I)
      RETURN
      END
      SUBROUTINE PPOINT
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      CALL PENSEL(15,0,0)
      CALL MOVTO2(XP(NN),YP(NN))
      CALL SYMBOL(7)
      CALL MOVBY2(-15.,-4.)
      CALL CHAINT(NN,5)
      CALL CHAMOD
      RETURN
      END
      SUBROUTINE BITCOM
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      CLOG=.FALSE.
      RLOG=.FALSE.
      NN=NN+1
      IF(A(1).LT.240.AND.A(1).GT.100.AND.A(2).LT.1920.AND.A(2).GT.1820)
     +GOTO 1000
      IF(A(1).LT.240.AND.A(1).GT.100.AND.A(2).LT.1780.AND.A(2).GT.1650)
     +GOTO 1001
      IF(A(1).LT.240.AND.A(1).GT.100.AND.A(2).LT.1600.AND.A(2).GT.1500)
     +GOTO 1002
      GOTO 1003
1000  CALL TNOU('Point accepted',14)
      GOTO 1003
1001  CALL TNOU('Point rejected, redefine node on BOTH drawings',46)
      NN=NN-1
      CLOG=.TRUE.
      RETURN
1002  CLOG=.FALSE.
      RLOG=.TRUE.
1003  RETURN
      END
      SUBROUTINE HELP1
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      WRITE(1,100)
100   FORMAT('MENU of Commands at this level is as follows :',/
     +'N     -     Generate node information',/
     +'A     -     Set bit-pad drawing dimensions (AXES)',/
     +'E     -     Generate element information',/
     +'P     -     Generate PAFBLOCKS information',/
     +'D     -     Draw all accepted information on screen',/
     +'S     -     Stop generating session, output data',/,
     +'Q     -     QUIT and dump all data generated to a file',/,
     +'R     -     RESUME session using data from previous dump',/,
     +'T     -     Change teminal type to tektronix',/,
     +)
      LOG=.TRUE.
      RETURN
      END
      SUBROUTINE EXIT1
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
1     CALL RNAM$A('Enter output file name',22,A$FUPP,OFIL,16)
      CALL SRCH$$(K$DELE,OFIL,16,0,K2,K1)
      CALL SRCH$$(K$WRIT,OFIL,16,1,K2,K1)
      CALL ERRPR$(K$IRTN,K1,0,0,0)
      IF(K1.NE.0) GOTO 1
C ----- WRITE A CONTROL MODULE
      WRITE(5,99) OFIL
99    FORMAT('CONTROL',/,'READ.FROM.',8A2,/,'DOUBLE',/,'CONTROL.END')
      WRITE(5,100)
C ------- WRITE STATEMENT MODIFIED TO ENSURE THAT E FORMAT
C ----- DOES NOT HAVE A SPACE AFTER THE E CHARACTER, AS PAFEC WILL NOT
C ----- READ AN E FORMAT WITH A SPACE.
      DO 105 I=1,NN
```

```
             J=0
             ENCODE(80,101,LINE) I,X(I),Y(I),Z(I)
       1000  J=J+1
             IF(J.GE.40) GOTO 105
             IF(LINE(J).EQ.'E ') LINE(J)='E+'
 ') LINE(J)='IF(LINE(J).EQ.'
') LINE(J)=' IF(LINE(J).EQ.'
             GOTO 1000
       105   CALL WTLIN$(1,LINE,40,K1)
       C ----- END OF NODE WRITE STATEMENT
             IF(EN.EQ.1) GOTO 1003
             WRITE(5,102)
             EN=EN-1
             DO 2 I=1,EN
             I2=TOPO(11,I)+2
       2     WRITE(5,103) I,(TOPO(J,I),J=1,I2)
       102   FORMAT('ELEMENTS',/,'NUMB ELEM PROP TOPO')
       103   FORMAT(I5,I10,9I5)
       C ----- PAFBLOCKS OUTPUT
       1003  IF(TOPOP(1,1).EQ.0) GOTO 110
             WRITE(5,111)
       111   FORMAT('PAFBLOCKS',/,'BLOC TYPE ELEM PROP N1 N2 N3 TOPO')
             PN=PN-1
             DO 3 I=1,PN
       3     WRITE(5,112)I,TOPOP(11,I),TOPOP(1,I),TOPOP(2,I),(TOPOP(J,I),J=12,
       1
            +4),(TOPOP(J1,I),J1=3,10)
       112   FORMAT(2I5,I10,12I5)
             WRITE(5,113)
       113   FORMAT('MESH',/,'REFE SPAC')
             DO 114 I=1,100
       114   IF(REFLOG(I)) WRITE(5,115) I,I
       115   FORMAT(2I5)
       110   CALL WTLIN$(1,'END.OF.DATA ',6,K1)
       1002  CALL SRCH$$(K$CLOS,0,0,1,K2,K1)
             CALL ERRPR$(K$NRTN,K1,0,0,0)
       100   FORMAT('NODES',/,'NODE X Y Z')
       101   FORMAT(I4,10X,E10.4,10X,E10.4,10X,E10.4)
             CALL DEVEND
             CALL EXIT
             END
             SUBROUTINE ELEM
       $INSERT TT>LJP>PAFCOM3
       $INSERT SYSCOM>A$KEYS
       $INSERT SYSCOM>KEYS.F
       $INSERT SYSCOM>ERRD.F
             LOG=.TRUE.
             J1=1
             IEN=2
       101   CALL TNOUA('Enter element type number : ',28)
             NEN=0
             READ(1,*,ERR=107) IEL
             IF(IEL.EQ.44200.OR.IEL.EQ.34200) NEN=4
             IF(IEL.EQ.36200) NEN=4
```

```
      IF(IEL.EQ.39200) NEN=4
      IF(IEL.EQ.39300) NEN=4
      IF(IEL.EQ.41320.OR.IEL.EQ.44100) NEN=3
      IF(IEL.EQ.36100) NEN=3
      IF(IEL.EQ.39100) NEN=3
      IF(IEL.EQ.44210) NEN=8
      IF(IEL.EQ.36610) NEN=8
      IF(IEL.EQ.36210) NEN=8
      IF(IEL.EQ.45210) NEN=8
      IF(IEL.EQ.43210) NEN=8
      IF(IEL.EQ.43310) NEN=3
      IF(IEL.EQ.39210) NEN=8
      IF(IEL.EQ.34000) NEN=2
      IF(IEL.EQ.44110) NEN=6
      IF(IEL.EQ.36110) NEN=6
      IF(IEL.EQ.45110) NEN=6
      IF(IEL.EQ.43110) NEN=6
      IF(IEL.EQ.39110) NEN=8
107   IF(NEN.EQ.0) CALL HELP3
      IF(NEN.EQ.0) GOTO 101
100   CALL CURDEF(' FRHEPBC*.')
      CALL TNOU('+-*/CD',6)
      CALL CURSOR(ICOM,CX,CY)
      IF(ICOM.EQ.2) RETURN
      IF(ICOM.EQ.3)GOTO 1002
      IF(ICOM.EQ.4) CALL HELP2
      IF(ICOM.EQ.4) GOTO 100
      IF(ICOM.EQ.5) GOTO 101
      IF(ICOM.EQ.6) GOTO 103
      IF(ICOM.EQ.7) GOTO 104
      IF(ICOM.EQ.8) IEN=IEN-1
      IF(ICOM.EQ.8) GOTO 105
      IF(ICOM.EQ.9) GOTO 1001
      ELOG=.FALSE.
      TOLE1=TOLE/10.
102   IF(ABS(CX-XP(J1)).LT.TOLE1.AND.ABS(CY-YP(J1)).LT.TOLE1) GOTO 1
105   J1=J1+1
      IF(J1.GT.NN) TOLE1=TOLE1*2.
      IF(J1.GT.NN) J1=1
      GOTO 102
1     CALL PENSEL(2,0,0)
      WRITE(1,995) J1
995   FORMAT('NODE CHOSEN = ',I3)
      CALL MOVTO2(XP(J1),YP(J1))
      CALL SYMBOL(8)
      CALL PENSEL(1,0,0)
      IEN=IEN+1
      TOPO(IEN,EN)=J1
      CALL TNOUA('',1)
      IF((IEN-2).GE.NEN)  GOTO 1003
      GOTO 100
1002  CALL SYMBOL(8)
      IEN=IEN-1
      GOTO 100
```

```
1001  CALL TNOU('Incorrect key pressed',21)
      GOTO 100
1003  CALL PLOTEL
      TOPO(1,EN)=IEL
      TOPO(2,EN)=IPROP
      TOPO(11,EN)=NEN
      EN=EN+1
      IEN=2
      ELOG=.TRUE.
      GOTO 100
103   CALL TNOUA('Enter property number : ',24)
      READ(1,*)IPROP
      GOTO 100
104   IEN=2
C     DELETEING ROUTINE FOR ELEMENT PICTURE
      IF(ELOG) EN=EN-1
      GOTO 100
      END
      SUBROUTINE HELP2
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      WRITE(1,1)
1     FORMAT('Menu of commands at ELEMENT level',//,
     +'<spacebar>  Select node at cursor',/,
     +'R     -     Reject last selected node',/,
     +'E     -     Redefine element type',/,
     +'F     -     Finish element generation, Return to Main Menu',/,
     +'P     -     Set element property number',/,
     +'B     -     Begin definition of present element again')
      RETURN
      END
      SUBROUTINE HELP3
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      WRITE(1,1)
1     FORMAT('Element types are as follows :',//,
     +'44200     -      4 noded bending and in-plane plate',/
     +'44210     -      8 noded     "          "         "',/,
     +'44100     -      3 noded     "          "         "',/,
     +'44110     -      6 noded     "          "         "',/,
     +'41320     -      3 noded hybrid stress plate bending',/,
     +'34000     -      2 noded simple beam element',/,
     +'34200     -      2 active+2 offset defining nodes, offset beam',/

     +'43210     -      8 Noded curved shell semi-loof element',/,
     +'43310     -      3 Noded curved beam semi-loof element',/,
     +'43110     -      6 Noded curved shell semi-loof element',/,
     +'45210     -      8 Noded thick shell element',/,
     +'45110     -      6 Noded thick shell element',/,
     +'39***     -      Series of elements for heat transfer calcs',/,
     +'36***     -     Series of isoparametric shell elements',/,
     +)
      RETURN
      END
      SUBROUTINE PLOTEL
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      CALL PENSEL(3,0,0)
      XC=0.0
      YC=0.0
      DO 1 I1=1,NEN
      I2=I1+2
      XC=XP(TOPO(I2,EN))+XC
1     YC=YP(TOPO(I2,EN))+YC
      XC=XC/NEN
      YC=YC/NEN
      CALL MOVTO2(XC,YC)
      CALL MOVBY2(-15.,-4.)
      CALL CHAINT(EN,3)
      IF(IEL.EQ.34200) CALL PL3420
      IF(IEL.EQ.34000) CALL PL3400
      IF(NEN.EQ.4) CALL PL4420
      IF(NEN.EQ.8) CALL PL4421
      IF(NEN.EQ.6) CALL PL4411
      IF(NEN.EQ.3) CALL PL4132
      CALL CHAMOD
      RETURN
      END
      SUBROUTINE PL3400
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      CALL PENSEL(4,0,0)
      CALL MOVTO2(XP(TOPO(3,EN)),YP(TOPO(3,EN)))
      CALL LINTO2(XP(TOPO(4,EN)),YP(TOPO(4,EN)))
      CALL PENSEL(3,0,0)
      RETURN
      END
      SUBROUTINE PL3420
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      CALL PENSEL(4,0,0)
      CALL MOVTO2(XP(TOPO(3,EN)),YP(TOPO(3,EN)))
      CALL LINTO2(XP(TOPO(5,EN)),YP(TOPO(5,EN)))
      CALL LINTO2(XP(TOPO(6,EN)),YP(TOPO(6,EN)))
      CALL LINTO2(XP(TOPO(4,EN)),YP(TOPO(4,EN)))
      CALL PENSEL(3,0,0)
      RETURN
      END
      SUBROUTINE PL4132
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      CALL MOVTO2(XP(TOPO(3,EN)),YP(TOPO(3,EN)))
      CALL LINTO2(XP(TOPO(4,EN)),YP(TOPO(4,EN)))
      CALL LINTO2(XP(TOPO(5,EN)),YP(TOPO(5,EN)))
      IF(IEL.EQ.43200) RETURN
      IF(TOPO(1,EN).EQ.43200) RETURN
      CALL LINTO2(XP(TOPO(3,EN)),YP(TOPO(3,EN)))
      RETURN
      END
      SUBROUTINE PL4411
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      CALL MOVTO2(XP(TOPO(3,EN)),YP(TOPO(3,EN)))
      CALL LINTO2(XP(TOPO(6,EN)),YP(TOPO(6,EN)))
      CALL LINTO2(XP(TOPO(4,EN)),YP(TOPO(4,EN)))
      CALL LINTO2(XP(TOPO(7,EN)),YP(TOPO(7,EN)))
      CALL LINTO2(XP(TOPO(5,EN)),YP(TOPO(5,EN)))
      CALL LINTO2(XP(TOPO(8,EN)),YP(TOPO(8,EN)))
      CALL LINTO2(XP(TOPO(3,EN)),YP(TOPO(3,EN)))
      RETURN
      END
      SUBROUTINE PL4420
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      CALL MOVTO2(XP(TOPO(3,EN)),YP(TOPO(3,EN)))
      CALL LINTO2(XP(TOPO(4,EN)),YP(TOPO(4,EN)))
      CALL LINTO2(XP(TOPO(6,EN)),YP(TOPO(6,EN)))
      CALL LINTO2(XP(TOPO(5,EN)),YP(TOPO(5,EN)))
      CALL LINTO2(XP(TOPO(3,EN)),YP(TOPO(3,EN)))
      RETURN
      END
      SUBROUTINE PL4421
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      CALL MOVTO2(XP(TOPO(3,EN)),YP(TOPO(3,EN)))
      CALL LINTO2(XP(TOPO(7,EN)),YP(TOPO(7,EN)))
      CALL LINTO2(XP(TOPO(4,EN)),YP(TOPO(4,EN)))
      CALL LINTO2(XP(TOPO(9,EN)),YP(TOPO(9,EN)))
      CALL LINTO2(XP(TOPO(6,EN)),YP(TOPO(6,EN)))
      CALL LINTO2(XP(TOPO(10,EN)),YP(TOPO(10,EN)))
      CALL LINTO2(XP(TOPO(5,EN)),YP(TOPO(5,EN)))
      CALL LINTO2(XP(TOPO(8,EN)),YP(TOPO(8,EN)))
      CALL LINTO2(XP(TOPO(3,EN)),YP(TOPO(3,EN)))
      RETURN
      END
      SUBROUTINE DRAW
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      LOG=.TRUE.
      CALL PICCLE
1000  ERLOG=.FALSE.
      CALL TNOUA('Enter viewing position : ',25)
      READ(1,*,ERR=999) VP
      IF(VP(1).EQ.0.0.AND.VP(2).EQ.0.0.AND.VP(3).EQ.0.0) ERLOG=.TRUE.
      IF(ERLOG) CALL TNOU('Cannot view from (0,0,0)',24)
      IF(ERLOG) GOTO 1000
      CALL PLSET
      IF(NN.EQ.0) RETURN
      NN1=NN
      DO 1 NN=1,NN1
      CALL PLOTCO
1     CALL PPOINT
      NN=NN1
      IF(EN.EQ.1) GOTO 3
      IEN1=EN-1
      DO 2 EN=1,IEN1
      IEL=TOPO(1,EN)
      NEN=TOPO(11,EN)
2     CALL PLOTEL
      EN=IEN1+1
3      IF(NP.EQ.1) RETURN
      INP=NP-1
      DO 4 NP=1,INP
      NPN=8
      IF(TOPOP(14,NP).EQ.0) NPN=6
      CALL PLOTPB
4     CALL PBENOE
      NP=INP+1
      CALL CHAMOD
      RETURN
999    CALL TNOU('Error in coordinates',20)
       GOTO 1000
      END
      SUBROUTINE PAFBL
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      LOG=.TRUE.
      J1=1
      IPN=2
101   CALL TNOUA('Enter element type number : ',28)
      NPN=0
      READ(1,*,ERR=107) IEL
108   CALL TNOUA('Enter pafblock type, 1 or 2 : ',30)
      READ(1,*,ERR=109) PT
```

```
       NPN=8
      IF(PT.EQ.2) NPN=6
      GOTO 100
107   CALL HELP3
      GOTO 101
100   CALL CURDEF(' FRHEPBC*.')
      CALL TNOU('+-*/CD',6)
      CALL CURSOR(ICOM,CX,CY)
      IF(ICOM.EQ.2) RETURN
      IF(ICOM.EQ.3)GOTO 1002
      IF(ICOM.EQ.4) CALL HELP2
      IF(ICOM.EQ.4) GOTO 100
      IF(ICOM.EQ.5) GOTO 101
      IF(ICOM.EQ.6) GOTO 103
      IF(ICOM.EQ.7) GOTO 104
      IF(ICOM.EQ.8) IPN=IPN-1
      IF(ICOM.EQ.8) GOTO 105
      IF(ICOM.EQ.9) GOTO 1001
      ELOG=.FALSE.
      TOLE1=TOLE*CXSCAL
102   IF(ABS(CX-XP(J1)).LT.TOLE1.AND.ABS(CY-YP(J1)).LT.TOLE1) GOTO 1
105   J1=J1+1
      IF(J1.GT.NN) TOLE1=TOLE1*2.
      IF(J1.GT.NN) J1=1
      GOTO 102
1     CALL PENSEL(2,0,0)
      WRITE(1,995) J1
995   FORMAT('NODE CHOSEN = ',I3)
      CALL MOVTO2(XP(J1),YP(J1))
      CALL SYMBOL(8)
      CALL PENSEL(15,0,0)
      IPN=IPN+1
      TOPOP(IPN,PN)=J1
      CALL TNOUA('',1)
      IF(IPN.GE.(NPN+2))  GOTO 1003
      GOTO 100
1002  CALL SYMBOL(8)
      IPN=IPN-1
      GOTO 100
1001  CALL TNOU('Incorrect key pressed',21)
      GOTO 100
1003  CALL PLOTPB
      CALL CHAMOD
      CALL TNOUA('Enter number of elements on MAUVE line : ',41)
      READ(1,*) TOPOP(12,PN)
      REFLOG(TOPOP(12,PN))=.TRUE.
      CALL TNOUA('Enter number of elements on TURQUOISE line : ',45)
      READ(1,*) TOPOP(13,PN)
      REFLOG(TOPOP(13,PN))=.TRUE.
      IF(NPN.EQ.6) CALL TNOUA('Enter number of elements on ***** line :
     +',41)
      IF(NPN.EQ.6) READ(1,*) TOPOP(14,PN)
      IF(NPN.EQ.6) REFLOG(TOPOP(14,PN))=.TRUE.
      CALL PBENOE
      TOPOP(1,PN)=IEL
      TOPOP(2,PN)=IPROP
      TOPOP(11,PN)=PT
      PN=PN+1
      IPN=2
      ELOG=.TRUE.
      GOTO 100
103   CALL TNOUA('Enter property number : ',24)
      READ(1,*)IPROP
      GOTO 100
104   IPN=2
C     DELETEING ROUTINE FOR ELEMENT PICTURE
      IF(ELOG) PN=PN-1
      GOTO 100
109   CALL TNOU('Pafblocks : 1= 8 noded rectangular',34)
      CALL TNOU('            2= 6 noded triangular.',34)
      GOTO 108
      END
      SUBROUTINE PLOTPB
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      XC=0.0
      YC=0.0
      DO 1 I1=1,NPN
      I2=I1+2
      XC=XP(TOPOP(I2,PN))+XC
1     YC=YP(TOPOP(I2,PN))+YC
      XC=XC/NPN
      YC=YC/NPN
      CALL MOVTO2(XC,YC)
      CALL PENSEL(5,0,0)
      CALL MOVBY2(-15.,-4.)
      CALL CHAINT(PN,3)
      IF(NPN.EQ.8) CALL PLPB1
      IF(NPN.EQ.6) CALL PLPB2
      RETURN
      END
      SUBROUTINE PBENOE
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      CALL CHASIZ(6.,4.)
      CALL PENSEL(5,0,0)
      CALL MOVTO2(XP(TOPOP(7,PN)),YP(TOPOP(7,PN)))
      I1=TOPOP(12,PN)
      CALL  CHAINT(I1,2)
      CALL PENSEL(6,0,0)
      CALL MOVTO2(XP(TOPOP(8,PN)),YP(TOPOP(8,PN)))
      I1=TOPOP(13,PN)
      CALL CHAINT(I1,2)
      CALL PENSEL(7,0,0)
      IF(NPN.EQ.6) CALL MOVTO2(XP(TOPOP(9,PN)),YP(TOPOP(9,PN)))
      I1=TOPOP(14,PN)
      IF(NPN.EQ.6) CALL CHAINT(I1,2)
      CALL CHASIZ(3.,2.)
      CALL CHAMOD
      RETURN
      END
      SUBROUTINE PLPB1
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      CALL MOVTO2(XP(TOPOP(3,PN)),YP(TOPOP(3,PN)))
      CALL PENSEL(5,0,0)
      CALL LINTO2(XP(TOPOP(7,PN)),YP(TOPOP(7,PN)))
      CALL LINTO2(XP(TOPOP(4,PN)),YP(TOPOP(4,PN)))
      CALL PENSEL(6,0,0)
      CALL LINTO2(XP(TOPOP(9,PN)),YP(TOPOP(9,PN)))
      CALL LINTO2(XP(TOPOP(6,PN)),YP(TOPOP(6,PN)))
      CALL PENSEL(5,0,0)
      CALL LINTO2(XP(TOPOP(10,PN)),YP(TOPOP(10,PN)))
      CALL LINTO2(XP(TOPOP(5,PN)),YP(TOPOP(5,PN)))
      CALL PENSEL(6,0,0)
      CALL LINTO2(XP(TOPOP(8,PN)),YP(TOPOP(8,PN)))
      CALL LINTO2(XP(TOPOP(3,PN)),YP(TOPOP(3,PN)))
      RETURN
      END
      SUBROUTINE PLPB2
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      CALL MOVTO2(XP(TOPOP(3,PN)),YP(TOPOP(3,PN)))
      CALL PENSEL(5,0,0)
      CALL LINTO2(XP(TOPOP(6,PN)),YP(TOPOP(6,PN)))
      CALL LINTO2(XP(TOPOP(4,PN)),YP(TOPOP(4,PN)))
      CALL PENSEL(6,0,0)
      CALL LINTO2(XP(TOPOP(7,PN)),YP(TOPOP(7,PN)))
      CALL LINTO2(XP(TOPOP(5,PN)),YP(TOPOP(5,PN)))
      CALL PENSEL(7,0,0)
      CALL LINTO2(XP(TOPOP(8,PN)),YP(TOPOP(8,PN)))
      CALL LINTO2(XP(TOPOP(3,PN)),YP(TOPOP(3,PN)))
      RETURN
      END
      SUBROUTINE PLSET
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      HEIGHT=0.0
      ANG=ATAN((CPY(5)-CPY(4))/(CPX(2)-CPX(1)))
      IF(VP(1).EQ.0.0.AND.VP(2).EQ.0.0) ALPHA=0.0
      IF(VP(1).EQ.0.0.AND.VP(2).EQ.0.0) GOTO 99
      ALPHA=ATAN2(VP(2),VP(1))
```

```
99     BETA2=SQRT(VP(1)*VP(1)+VP(2)*VP(2))
       BETA=ATAN2(VP(3),BETA2)
       IF(ABS(BETA).GT.1.55) GOTO 100
       XPMIN=(CPX(2)-CPX(1))*SIN(ALPHA)*1.1
       WIDTH=XPMIN+(CPY(5)-CPY(4))*COS(ALPHA)
       HEIGHT=((CPZ(3)-CPZ(1)+CPZ(6)-CPZ(4))/2.)*COS(BETA)
       YPMIN=SQRT((CPX(2)-CPX(1))**2+(CPY(5)-CPY(4))**2)*COS(ANG-ALPHA)*
S
      +IN(BETA)/(COS(BETA)*.9)
       HEIGHT=HEIGHT+YPMIN
       GOTO 101
100    HEIGHT=CPY(5)-CPY(4)
       WIDTH=CPX(2)-CPX(1)
       YPMIN=0.0
       XPMIN=0.0
       GOTO 101
101    PSCAL1=220./HEIGHT
       PSCAL2=250./WIDTH
       PSCAL=AMIN1(PSCAL1,PSCAL2)
       RETURN
       END
       SUBROUTINE PLOTCO
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
       XN=X(NN)-CPY(1)
       YN=Y(NN)-CPY(4)
       IF(ABS(BETA).GT.1.55) GOTO 3
       XP(NN)=(YN*COS(ALPHA)+XN*SIN(ALPHA)+XPMIN)*PSCAL
       IF(YN.EQ.0.0.AND.XN.EQ.0.0) GOTO 2
1      YP(NN)=(Z(NN)*COS(BETA)-SQRT(XN**2+YN**2)*SIN(BETA)*COS(ATAN2(YN,
X
      +N)-ALPHA)+YPMIN)*PSCAL
       RETURN
2      Y(NN)=(Z(NN)*COS(BETA)+YPMIN)*PSCAL
       RETURN
3      XP(NN)=XN*PSCAL
       YP(NN)=YN*PSCAL
       RETURN
       END
       SUBROUTINE EXIT2
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
       CALL RNAM$A('Enter DUMP file name',20,A$FUPP,OFIL,16)
       CALL SRCH$$(K$WRIT,OFIL,16,1,K2,K1)
       CALL ERRPR$(K$NRTN,K1,0,0,0)
       CALL PRWF$$(K$WRIT,1,LOC(A),14207,000000,IRNW,K1)
       CALL TNOU('DATA WRITTEN',12)
       CALL ERRPR$(K$NRTN,K1,0,0,0)
       CALL SRCH$$(K$CLOS,OFIL,16,1,K2,K1)
       CALL EXIT
      END
      SUBROUTINE SAVE
$INSERT TT>LJP>PAFCOM3
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F
      CALL RNAM$A('Enter name of DUMPed file',25,A$FUPP,OFIL,16)
      CALL SRCH$$(K$READ,OFIL,16,1,K2,K1)
      CALL ERRPR$(K$NRTN,K1,0,0,0)
      CALL PRWF$$(K$READ,1,LOC(A),14207,000000,IRNW,K1)
      CALL ERRPR$(K$NRTN,K1,0,0,0)
      CALL SRCH$$(K$CLOS,OFIL,16,1,K2,K1)
      LOG=.TRUE.
      LAXIS=.FALSE.
      RETURN
      END
```

Appendix 3 - Graphics Routines.

Graphics routines to drive Matrox ALT-512 graphics board.

This facility was considered necessary to allow the display of data while the experiment was under way. The Matrox ALT-512 graphics board is a very simple display board made up of a 512x256 dot display on two planes, each of 256x256 dots. Each dot may be addressed through X and Y address registers through 6 Z80 ports (8-13). Supplied with the board were a series of graphics routines written by Universal Graphics Interpreters, however these were fairly difficult to combine with a FORTRAN program, requiring the procedure given below:

A series of subroutines are required which put the necessary plotting commands for the UGI routines into an array and a machine code routine to tell the graphics software where to find the data by inserting the address into a specific location and then calling the UGI routines at their start address.
The Compiling sequence and loading sequence is then as follows:

1) Compile the FORTRAN Program	f80 =prog
2) Load at a high address (above UGI)	l80 /p:1d00,prog,lib
3) Make a map to find start of $main	/m
4) Exit from loader	/e
5) Enter ddt	ddt prog.com
6) Insert UGI	iugimx512.obj
	r
7) Insert jump to $main	s100
	c3
	low byte
	high byte of $main
	.
8) Leave ddt	g0
9)Saveprogram	SAVEPROG.COM PAGES

The map gives the number of pages.

Appendix 3 - Graphics Routines.

These routines only produce a 256x256 grid (since they write the identical information to both planes) and it was decided that this resolution was insufficient. The author therefore had to write a series of Machine code and FORTRAN routines which could be loaded and called more conveniently.

For speed of operation it is necessary to use machine code for the basic line drawing routines. The algorithm used for the line drawing was borrowed from the SINCLAIR ZX81 manual where it is given as a 50 line BASIC program. Considerable effort was required to modify this to machine code, since no division or multplication can be done. The machine code program shows its origins in a BASIC form as each statement from the original program is simply replaced by a block of assembler instructions. All of the arithmetic could be performed direct by the Z80 instruction set since all the numbers are single byte values (0-255) due to the way the Matrox board is configured. The program can be called directly from FORTRAN (though a scaling subroutine would normally be used fo REAL dimensions) by CALL LIN(IX,IY) in which case a line is drawn from the current cursor position (X,Y) to the new position (IX,IY) - $0<IX<512$ and $0<IY<256$. This routine would not normally be used unless high speed operation was required. The FORTRAN routine LINTO2(X,Y) scales real values in mm to those reqired for the Matrox board.

The routine MOV(IX,IY) simply places the new IX and IY values in the cursor registers.

The routine COLOUR(N) places a colour code number in the COLOR register of the program, this facility is useful if extra colour boards are added at a future date.

The routine CHAR(I) writes the ASCII character I on the graphics screen at the current cursor position. The routine was written to draw the full ASCII character set, both upper and lower case (it may be useful to add Greek at a later stage). The characters are formed on a 5x7 matrix of dots and are identical to those on the Tuscan internal VDU. The data block alone for the characters occupy 480bytes of memory.

The routine GRDUMP sends a bit map to the EPSON MX100 printer for permanent graphical records. The use of the author's own printer driver routine was required as the Tuscan monitor will not output an ASCII 12 (form feed) character without adding extra line feed characters which distort the bit image print. The author's routine simply transfers the characters piecemeal and causes no problems.

To drive these routines more conveniently from FORTRAN a series of short FORTRAN routines were written which perform the neccessary scaling to convert input in mm to paper dimensions. The normally used routines are described below. Their relocatable machine code is stored in the file LJP.REL and to load then into a program simply use a command in the following form after compiling your program:

```
A> L80 prog,LJP/S,FORLIB/S,prog/N/E
```

L80 is the Microsoft linking loader.

Graphical subroutines.

FORTRAN callable Graphics subroutines. All dimensions are in mm on the printed graph, not on the screen which is considerably taller.

`CALL MATROX`

This routine should be called before all other graphics routines to set up the graphics board.

`CALL MOVTO2(X,Y)`

Move the cursor to the position X,Y.

`CALL LINTO2(X,Y)`

Draw a line from current cursor position to X,Y.

`CALL CHAR('I')`

Draw a single character 'I' at the cursor position.

`CALL CHAARR(IARRAY,N)`

Draw a string of N characters from IARRAY, One character per byte (i.e. A2 if IARRAY is INTEGER*2). The cursor is left at the end of the last character.

`CALL PICCLE`

Picture Clear.

`CALL COLOUR(N)`

Set the colour for all subsequent graphical output.

For black and white an odd number = white

an even number = black

`CALL GRDUMP`

Dump the picture to the printer.

Appendix 3 - Graphics Routines.

```
          CALL AXIS(XLEN,YLEN)
```

Draw a set of axes with lengths as given.

```
          CALL SCAL(IXTIKS,IYTIKS)
```

Put major ticks on the axes. The two values give the number of ticks to be equally spaced on each axis.

```
          CALL AXISCA(XBEG,XEND,YBEG,YEND)
```

Write scales on the axes at the ticks starting and ending at the values given.

```
          CALL GRALIN(X,Y,N)
```

Draw a graph through n points (X,Y) on the graph axes

```
          CALL GRASYM(X,Y,N,'*')
```

Plot N points X,Y on the graph axes using symbol '*'.

```
          CALL GRAF(X,Y,N)
```

Draw a set of scaled axes and a line graph for the N data points X,Y.

MICRO GRAPHICS ROUTINES - FORTRAN 80

```
        subroutine graf(x,y,n)
        dimension x(n),y(n)
        xmax=x(1)
        xmin=x(1)
        ymax=y(1)
        ymin=y(1)
        do 1 i=2,n
        if(x(i).gt.xmax) xmax=x(i)
        if(x(i).lt.xmin) xmin=x(i)
        if(y(i).gt.ymax) ymax=y(i)
        if(y(i).lt.ymin) ymin=y(i)
1       continue
        call piccle
        call axis(170.0,70.0)
        call scal(10,10)
        call axisca(xmin,xmax,ymin,ymax)
        call gralin(x,y,n)
        return
        end
c
c
c
        subroutine grasym(x,y,n,m)
        dimension x(n),y(n)
        include com.gra
        sx=lenx/xran
        sy=leny/yran
        do 1 i=1,n
        xco=(x(i)-xbeg)*sx+19.0
        yco=(y(i)-ybeg)*sy+10.5
        call movto2(xco,yco)
1       call char(m)
        return
        end
c
c
c
        subroutine axisca(a,b,c,d)
        include com.gra
        xbeg=a
        xend=b
        ybeg=c
        yend=d
        ixexp=0
        iyexp=0
        xran=xend-xbeg
        if(xran.ne.0.0) goto 3
        xbeg=1.1*xbeg
        xend=1.1*xend
        xran=xend-xbeg
        if(xran.eq.0.0) return
3       yran=yend-ybeg
        if(yran.ne.0.0) goto 4
        ybeg=1.1*ybeg
        yend=1.1*yend
        yran=yend-ybeg
        if(yran.eq.0.0) return
4       xbeg1=abs(xbeg)
        if(abs(xend).gt.xbeg1) xbeg1=abs(xend)
        if(xbeg1.lt.1.) goto 100
101     if(xbeg1.gt.1.) ixexp=ixexp+1
        if(xbeg1.gt.1.) xbeg1=xbeg1/10.
        if(xbeg1.gt.1.) goto 101
112     ybeg1=abs(ybeg)
        if(abs(yend).gt.ybeg1) ybeg1=abs(yend)
        if(ybeg1.lt.1.) goto 110
111     if(ybeg1.gt.1.) iyexp=iyexp+1
        if(ybeg1.gt.1.) ybeg1=ybeg1/10.
        if(ybeg1.gt.1.) goto 111
        goto 120
100     if(xbeg1.lt.0.1) ixexp=ixexp-1
        if(xbeg1.lt.0.1) xbeg1=xbeg1*10.
        if(xbeg1.lt.0.1) goto 100
        goto 112
110     if(ybeg1.lt.0.1) iyexp=iyexp-1
        if(ybeg1.lt.0.1) ybeg1=ybeg1*10.
        if(ybeg1.lt.0.1) goto 110
120     xran1=xran/10.0**ixexp
        yran1=yran/10.0**iyexp
        xbeg1=xbeg/10.0**ixexp
        ybeg1=ybeg/10.0**iyexp
        do 1 i=1,ixd
        xval=xbeg1+(xran1*i)/ixd
        encode(lab,150) xval
150     format(f5.3)
        xp=i*xdiv+15.
        call movto2(xp,7.0)
1       call chaarr(lab,5)
        call movto2(xp+14.,7.0)
        call chaarr('*10^',4)
        encode(lab,151) ixexp
151     format(i2)
        call movto2(xp+24.,9.0)
        call chaarr(lab,2)
        do 2 i=1,iyd
        yp=i*ydiv+9.
        call movto2(3.,yp)
        yval=ybeg1+(yran1*i)/iyd
        encode(lab,150) yval
2       call chaarr(lab,5)
        call movto2(10.,yp+5.)
        call chaarr('*10^',4)
        call movto2(20.0,yp+7.)
        encode(lab,151) iyexp
        call chaarr(lab,2)
        return
        end
c
c
c
```

```
      subroutine gralin(x,y,n)
      dimension x(n),y(n)
      include com.gra
      sx=lenx/xran
      sy=leny/yran
      xco=(x(1)-xbeg)*sx+20.0
      yco=(y(1)-ybeg)*sy+12.0
      call movto2(xco,yco)
      do 1 i=2,n
      xco=(x(i)-xbeg)*sx+20.0
      yco=(y(i)-ybeg)*sy+12.0
1     call linto2(xco,yco)
      return
      end
c
c
c
      subroutine axis(a,b)
      include com.gra
      lenx=a
      leny=b
      call movto2(20.,12.)
      call linto2(lenx+20.,12.)
      call movto2(20.,12.)
      call linto2(20.,leny+12.)
      return
      end
c
c
c
      subroutine scal(m,n)
      include com.gra
      ixd=m
      iyd=n
      xdiv=lenx/ixd
      ydiv=leny/iyd
      do 1 i=1,ixd
      xp=i*xdiv+20.
      call movto2(xp,12.)
1     call linto2(xp,10.)
      do 2 i=1,iyd
      yp=i*ydiv+12.
      call movto2(20.,yp)
2     call linto2(18.,yp)
      mdivx=20
      if(xdiv.lt.50.) mdivx=10
      if(xdiv.lt.20.) mdivx=5
      if(xdiv.lt.10.) mdivx=2
      if(xdiv.lt.5.) goto 10
      nsdivx=mdivx*ixd
      sxdiv=lenx/nsdivx
      do 3 i=1,nsdivx
      xp=i*sxdiv+20.
      call movto2(xp,12.0)
3     call linto2(xp,11.0)
```

```
10      mdivy=20
        if(ydiv.lt.50.) mdivy=10
        if(ydiv.lt.20.) mdivy=5
        if(ydiv.lt.10.) mdivy=2
        if(ydiv.lt.5.) return
        nsdivy=mdivy*iyd
        sydiv=leny/nsdivy
        do 4 i=1,nsdivy
        yp=i*sydiv+12.0
        call movto2(20.0,yp)
4       call linto2(19.,yp)
        return
        end
c
c
        subroutine matrox
        call piccle
        call colour(1)
        call out(13,12)
        return
        end
c
c
        subroutine linto2(x,y)
        include com.gra
        ix=x*2.359
        iy=y*2.844
        if(ix.gt.511.or.iy.gt.255) return
        call lin(ix,iy)
        xcord=x
        ycord=y
        return
        end
c
c
        subroutine movto2(x,y)
        include com.gra
        ix=x*2.359
        iy=y*2.844
        if(ix.gt.511.or.iy.gt.255) return
        call mov(ix,iy)
        xcord=x
        ycord=y
        return
        end
c
c
        subroutine chaarr(str,n)
        byte str(n)
        include com.gra
        do 1 i=1,n
1       call char(str(i))
        return
        end
c
```

MICRO GRAPHICS ROUTINES - Z80 MACHINE CODE

```
                .z80
lin::           jp      lin1
; Matrox ALT-512  routine for drawing straight lines
; on a 512 x 256 grid, with colour.
; Algorithm based on that in SINCLAIR ZX81 Manual
        color:  db      7
        x:      db      0,0
        y:      db      0,0
        xn:     ds      2
        yn:     ds      2
        up:     ds      2
        acr:    ds      2
        npltd:  ds      2
        s:      ds      2
        d1x:    ds      2
        d1y:    ds      2
        d2x:    ds      2
        d2y:    ds      2
        m0:     ds      2
        n:      ds      2
;
;
        lin1:   push    hl
                push    de
                ld      c,(hl)          ;read x value
                inc     hl
                ld      b,(hl)
                ld      (xn),bc         ;save x in xn
                push    de
                pop     hl
                ld      c,(hl)          ;read y value
                inc     hl
                ld      b,(hl)
                ld      (yn),bc         ;save y in yn
; Set npltd (no of points plotted) to minus one
                ld      hl,-1
                ld      (npltd),hl
; Calculate the number of steps across
;
                ld      hl,(xn)
                ld      de,(x)
                and     a
                sbc     hl,de
                ld      (acr),hl
;set diagonal x
;
                bit     7,h
                jp      nz,diag1
                ld      hl,1
                jp      diag2
        diag1:  ld      hl,-1
        diag2:  ld      (d1x),hl
                ld      (d2x),hl
                ld      hl,0
                ld      (d2y),hl
```

```
; Calculate number of steps up
;
                ld      hl,(yn)
                ld      de,(y)
                and     a
                sbc     hl,de
                ld      (up),hl
;set diagonal y
;
                bit     7,h
                jp      nz,diag3
                ld      hl,1
                jp      diag4
        diag3:  ld      hl,-1
        diag4:  ld      (d1y),hl
; Set m=abs v
                ld      hl,(acr)
                bit     7,h
                call    nz,twoc
                ld      (m0),hl
; Set n=abs u
                ld      hl,(up)
                bit     7,h
                call    nz,twoc
                ld      (n),hl
;If m>n go and plot
;
                ld      de,(m0)
                and     a
                sbc     hl,de
                jp      m,mset
;reset diagonal step
                ld      hl,0
                ld      (d2x),hl
                ld      hl,(d1y)
                ld      (d2y),hl
                ld      de,(m0)
                ld      hl,(n)
                ld      (m0),hl
                ld      (n),de
;set value of s: to m/2
;
        mset:   ld      hl,(m0)
                srl     h
                rr      l
                ld      (s),hl
;set plane for plot of point
;
        plot:   ld      hl,(x)
                bit     0,l
                jp      z,pla
        plb:    ld      a,1
                jp      plane
        pla:    ld      a,0
        plane:  out     (12),a
```

```
;set x coordinate
                srl     h
                rr      l
                ld      a,l
;output x coordinate
                out     (9),a
                ld      a,(y)
                cpl
;output y coordinate
                out     (10),a
                ld      a,(color)
;output point in colour
                out     (8),a
; any more points?
;
                ld      hl,(npltd)
                inc     hl
                ld      (npltd),hl
                ld      de,(m0)
                and     a
                sbc     hl,de
                jp      m,cont1
        done:   pop     de
                pop     hl
                ret
; Calculate next x and y coordinate
;
        cont1:
; s=s+n
                ld      de,(s)
                ld      hl,(n)
                add     hl,de
                ld      (s),hl
; compare s and m (m0)
                ld      de,(m0)
                and     a
                sbc     hl,de
                jp      m,strgt
;diagonal increment
; s=s-m
        diag:   ld      hl,(s)
                ld      de,(m0)
                and     a
                sbc     hl,de
                ld      (s),hl
; a=a+d1x
                ld      hl,(x)
                ld      de,(d1x)
                add     hl,de
                ld      (x),hl
;b=b+d1y
                ld      hl,(y)
                ld      de,(d1y)
                add     hl,de
                ld      (y),hl
                jp      plot
;straight increment
        strgt:  ld      hl,(x)
                ld      de,(d2x)
                add     hl,de
                ld      (x),hl
                ld      hl,(y)
                ld      de,(d2y)
                add     hl,de
                ld      (y),hl
                jp      plot
        twoc:   ld      a,h
                cpl
                ld      h,a
                ld      a,l
                cpl
                ld      l,a
                inc     hl
                ret
; Routine to set colour of subsequent lines
;
        Colour::
                push    hl
                ld      a,(hl)  ; Read colour byte
                ld      (color),a       ; Save in color register
                pop     hl
                ret
;Routine to clear both picturs planes
;
        piccle::
                ld      a,1
                out     (12),a  ; Switch to plane B
                ld      a,0
                out     (11),a  ; Clear instruction
                call    clrst
                out     (12),a  ; Switch to plane A
                out     (11),a  ; Clear instruction
                call    clrst
                ret
        clrst:  push    af
        loop1p: in      a,(9)   ; Read Matrox flag reg.
                bit     3,a     ; Check clear status
                jp      z,loop1p        ; Loop if not ready
                pop     af
                ret             ; Return if ready
; Routine to move cursor position
;
        mov::   push    hl
                push    de
                ld      c,(hl)
                inc     hl
                ld      b,(hl)  ; Read in new x pos.
                ld      (x),bc  ; Save in x reg.
                push    de
                pop     hl
                ld      c,(hl)  ; Read in new y pos.
                inc     hl
                ld      b,(hl)
                ld      (y),bc  ; Save in y reg
                pop     de
                pop     hl
                ret
;       Routine to read bit map of one line from
;       graphics board type MATROX-512 to printer
;       type EPSON MX-100III
grdump::
                jp      beg
; Escape sequence for bit image printing
esc:            db      27,65,8,27,75,0,2
yval:           db      0
col:            ds      1
beg:            ld      a,0
esclp1:         ld      hl,esc  ; Output escape sequence
esclp:          ld      e,(hl)
                call    lpt
                inc     hl
                inc     a
                cp      7
                jp      nz,esclp
                ld      b,0
                ld      e,0
loop3:          ld      a,(yval)
                ld      c,a
                ld      a,0
                out     (12),a  ; Switch to plane A
loop11:         ld      a,b
                out     (9),a   ; output x coord
                ld      a,c
                out     (10),a  ; Output y coord
                in      a,(8)   ; Read bit
                ld      d,a
                res     0,d
                xor     d
                add     a,e     ; Add 2*old dot pattern
                add     a,e     ; i.e. Shift to the left
                ld      e,a     ; save new pattern in e reg.
                inc     c       ; Increment y coord
                ld      a,c
                ld      hl,yval
                ld      d,(hl)
                sub     d
                cp      8       ; Chech if 8th bit?
                jp      nz,loop11
                call    lpt     ; If so then print
                ld      e,0
                ld      a,1
                out     (12),a  ; Switch to plane B
; Same as for plane A ----------------------------
                ld      a,(yval)
                ld      c,a
```

```
loop2:          ld      a,b
                out     (9),a
                ld      a,c
                out     (10),a
                in      a,(8)
                ld      d,a
                res     0,d
                xor     d
                add     a,e
                add     a,e
                ld      e,a
                inc     c
                ld      a,c
                ld      hl,yval
                ld      d,(hl)
                sub     d
                cp      8
                jp      nz,loop2
                call    lpt
; ------------------------------------------------
                ld      e,0
                inc     b       ; Increment x coord
                ld      a,b
                cp      0       ; x=256?
                jp      nz,loop3
                ld      e,10
                call    lpt     ; If so finish line
                ld      a,(yval)
                add     a,8
                ld      (yval),a        ; Next line (yval+8)
                cp      0       ; Last line?
                jp      nz,beg
                ld      e,27    ; If so reset printer
                call    lpt
                ld      e,65
                call    lpt
                ld      e,12
                call lpt
                ret
; Printer driver routine, required since
; CP/M adds extra characters. Character
; is transfered in e register.
        lpt:    push    af
        lpt1:   in      a,(7)
                bit     0,a     ; Check if printer ready
                jp      nz,lpt1 ; if not loop
                ld      a,e
                out     (3),a   ; Output character to printer
                ld      a,0
                out     (7),a   ; Send strobe pulse
                nop
                nop
                ld      a,1     ; End strobe pulse
                out     (7),a
                pop     af
                ret

;routine to call universal graphics programs
;
        Graph::
        push    hl
        ld      (0359h),hl
        call    03c8h
        pop     hl
; check error register
                ld      a,(035ah)
                cp      0
                ret     z
error:          db      'graphical error$'
                ld      de,error
                ld      c,9
                call    5
                ret
; Routine for character generation ASCII 2Ah to 3Dh
;
        char::  jp      chabeg
        dat:    ds      1               ; Holds byte to print
        chapos: ds      2               ; Holds pos. of byte in table
        symb:
                db      00h,00h,00h,00h,00h     ;
                db      00h,00h,5fh,00h,00h     ;!
                db      00h,07h,00h,07h,00h     ;"
                db      14h,7fh,14h,7fh,14h     ;#
                db      24h,2ah,7fh,2ah,12h     ;$
                db      23h,13h,08h,64h,62h     ;%
                db      36h,49h,56h,20h,50h     ;&
                db      00h,04h,02h,01h,00h     ;'
                db      1ch,22h,41h,00h,00h     ;(
                db      00h,00h,41h,22h,1ch     ;)
                db      2ah,1ch,7fh,1ch,2ah     ;*
                db      08h,08h,3eh,08h,08h     ;+
                db      00h,80h,60h,00h,00h     ;,
                db      08h,08h,08h,08h,08h     ;-
                db      00h,00h,40h,00h,00h     ;.
                db      20h,10h,08h,04h,02h     ;/
                db      3eh,51h,49h,45h,3eh     ;0
                db      00h,42h,7fh,40h,00h     ;1
                db      62h,51h,49h,49h,46h     ;2
                db      21h,41h,49h,4dh,33h     ;3
                db      18h,14h,12h,7fh,10h     ;4
                db      27h,45h,45h,45h,39h     ;5
                db      3ch,4ah,49h,49h,31h     ;6
                db      01h,71h,09h,05h,03h     ;7
                db      36h,49h,49h,49h,36h     ;8
                db      46h,49h,49h,29h,1eh     ;9
                db      00h,00h,14h,00h,00h     ;:
                db      00h,40h,34h,00h,00h     ;;
                db      08h,14h,22h,41h,00h     ;<
                db      14h,14h,14h,14h,14h     ;=
                db      00h,41h,22h,14h,08h     ;>
db          02h,01h,59h,05h,02h          ;?
db          3eh,41h,5dh,59h,4eh          ;@
db          7ch,12h,11h,12h,7ch          ;A
db          7fh,49h,49h,49h,36h          ;B
db          3eh,41h,41h,41h,22h          ;C
db          7fh,41h,41h,41h,3eh          ;D
db          7fh,49h,49h,49h,41h          ;E
db          7fh,09h,09h,09h,01h          ;F
db          3eh,41h,41h,51h,71h          ;G
db          7fh,08h,08h,08h,7fh          ;H
db          00h,41h,7fh,41h,00h          ;I
db          20h,40h,40h,40h,3fh          ;J
db          7fh,08h,14h,22h,41h          ;K
db          7fh,40h,40h,40h,40h          ;L
db          7fh,02h,0ch,02h,7fh          ;M
db          7fh,04h,08h,10h,7fh          ;N
db          3eh,41h,41h,41h,3eh          ;O
db          7fh,09h,09h,09h,06h          ;P
db          3eh,41h,51h,21h,5eh          ;Q
db          7fh,09h,19h,29h,46h          ;R
db          26h,49h,49h,49h,32h          ;S
db          01h,01h,7fh,01h,01h          ;T
db          3fh,40h,40h,40h,3fh          ;U
db          1fh,20h,40h,20h,1fh          ;V
db          7fh,20h,10h,20h,7fh          ;W
db          63h,14h,08h,14h,63h          ;X
db          03h,04h,78h,04h,03h          ;Y
db          61h,51h,49h,45h,43h          ;Z
db          7fh,41h,41h,41h,41h          ;[
db          02h,04h,08h,10h,20h          ;/
db          41h,41h,41h,41h,7fh          ;]
db          04h,02h,7fh,02h,04h          ;^
db          40h,40h,40h,40h,40h          ;
db          00h,1ch,1ch,1ch,00h          ;
db          24h,54h,54h,54h,68h          ;a
db          7bh,44h,44h,44h,38h          ;b
db          38h,44h,44h,44h,44h          ;c
db          38h,44h,44h,44h,7bh          ;d
db          38h,54h,54h,54h,48h          ;e
db          04h,7eh,05h,01h,02h          ;f
db          48h,54h,54h,54h,2ah          ;g
db          7bh,04h,04h,04h,78h          ;h
db          00h,44h,7dh,40h,00h          ;i
db          20h,40h,40h,40h,3dh          ;j
db          7fh,10h,18h,24h,40h          ;k
db          00h,41h,7fh,40h,00h          ;l
db          78h,04h,78h,04h,78h          ;m
db          78h,04h,04h,04h,78h          ;n
db          38h,44h,44h,44h,38h          ;o
db          6ch,14h,14h,14h,08h          ;p
db          38h,44h,54h,24h,58h          ;q
db          04h,78h,04h,04h,08h          ;r
db          48h,54h,54h,54h,24h          ;s
db          04h,3fh,44h,40h,20h          ;t
db          3ch,40h,40h,40h,3ch          ;u
```

```
                db      1ch,20h,40h,20h,1ch      ;v
                db      3ch,40h,38h,40h,3ch      ;w
                db      44h,28h,10h,28h,44h      ;x
                db      4ch,50h,50h,50h,2ch      ;y
                db      44h,64h,54h,4ch,44h      ;z
                db      08h,08h,36h,41h,41h      ;{
                db      00h,00h,77h,00h,00h      ;
                db      41h,41h,36h,08h,08h      ;}
                db      08h,04h,08h,10h,08h      ;~
        chabeg:
                ld      a,(hl)  ;a=ASCII char requd
; Is character within range 20h-7eh?
                cp      07fh
                ret     p       ; Return if too high
                ld      a,(hl)
                cp      020h
; Change 20h to lowest ASCII character defined at symb
                ret     m       ; Return if too low
                push    hl
; find position in symbol table chapos=(a-020h)*5
; Put in a register
                sub     020h
                ld      b,0
                ld      e,a
                sla     a       ; *2
                sla     a       ; *4
                jp      nc,noca1
                inc     b       ; b reg. holds carry
        noca1:  add     a,e     ; *5
                jp      nc,noca2
                inc     b
        noca2:  ld      de,symb
; Add to adress of symb
                add     a,e
                jp      nc,noca ;check for carry into d
                inc     d
        noca:   ld      e,a
; add carry to d reg from multiply by five
                ld      a,d
                add     a,b
                ld      d,a
                ld      bc,00h
        nextc:  ld      a,(de)  ; Next column
                ld      (dat),a ; Save byte in dat
                ld      (chapos),de      ; Save table position
        nextr:                  ; next row of character
                ld      a,(dat)
                sla     a       ; Move next bit to 0
                ld      (dat),a
                jp      nc,blank
;set x coordinate and plane
                ld      hl,(x)
                bit     0,l     ; Check which plane
                jp      z,plna
        plnb:   ld      a,1     ; Plane B
        jp      plne
plna:   ld      a,0     ; Plane A
plne:   out     (12),a  ; Swith on plane
        srl     h
        rr      l       ; Divide x coord by 2
        ld      a,l
        out     (9),a   ; Output x coord
        ld      a,(y)
        cpl
        out     (10),a  ; output y coord
        ld      a,(color)
        out     (8),a   ; Output dot
blank:  inc     b
        ld      hl,(y)
        inc     hl      ; Increment y coord
        ld      (y),hl
        ld      a,b
        cp      8       ; Last byte of row?
        jp      nz,nextr
        ld      b,0
        inc     c
        ld      hl,(y)
        ld      de,-8   ; Reset y coord
        add     hl,de
        ld      (y),hl
        ld      hl,(x)
        inc     hl      ; Increment x coord
        ld      (x),hl
        ld      de,(chapos)
        inc     de      ; set chapos to next byte
        ld      (chapos),de
        ld      a,c
        cp      5       ; Check if last byte
        jp      nz,nextc
        ld      hl,(x)
        inc     hl      ; set x coord ready for next char
        ld      (x),hl
        pop     hl
        ret
        end
```

Appendix 4.

Routines to drive Analogue input and output devices from FORTRAN programs.

CDC AD-100-2 ADC.

This device is a 12-bit 16 channel Analogue to Digital Converter with a programmable gain from 1 to 1024 in powers of 2. This gives a full scale deflection of between 10V and 10mV (a maximum resolution of 5μV). The circuit consists of a sample and hold amplifier, a programmable gain amplifier, the ADC (12-bit) and all the necessary multiplexing and decoding circuitry.

The board is operated through 4 Z80 ports (A0h to A3h). The purpose of each is explained in the program listings.

The board is presently wired to give 8 differential inputs. It is only necessary to use single ended inputs to read the potentiometers, but the load cell requires a differential input so that all the channels have to be differential. The routine for driving this board is written in machine code for simplicity. It may be called directly from a FORTRAN program.

The call is as follows:

```
CALL ADREAD(ICHAN,IGADC,IVAL)
```

ICHAN is the channel number and is an INTEGER or a BYTE value in the range 0-7.

IGADC is the gain specifier and is an INTEGER or BYTE value

IVAL is the returned analogue voltage reading. (8 readings added together) and is an INTEGER*2 in the range -16384 to 16376.

Appendix 4 - External control routines.

To obtain the required value for IGADC, use the routine:

CALL GAINAD(IGAIN,IGSET,IGADC)

IGAIN	is the required gain	e.g. 853
IGSET	is the gain actually set	e.g. 512
IGADC	is the value for the amplifier	e.g. 13

All values are INTEGER*2. Note that the amplifier gain can only be set to powers of 2.

Cromemco 7A+D

It was initially intended to use a Transam I/O-3 Digital to Analogue converter, capable of delivering a current of 10mA from its sample and hold amplifiers. This board was not delivered in time for the experiment so that the Cromemco board had to be substituted. The specification for this board is a maximum output current of 1.5mA. It should therefore operate across a load of 2KΩ at 2.5V, but would in fact only operate into 6KΩ.

The Cromemco board has 7 8-bit DAC channels all indevidually adressed by a single Z80 port (19h-1Fh). This makes programming simpler but uses a large number of ports. The following operating routine is written in FORTRAN:

CALL DAC(ICHAN,VOLTS)

ICHAN is the channel number 1-8 and is INTEGER*2.

VOLTS is a REAL*4 number in the range -2.56 to 2.56. This value is rounded to the next lowest possible voltage on return.

If VOLTS were sent as 1.0325V the analogue output would be set to 1.02V and VOLTS would be returned as 1.02V.

Appendix 4 - External control routines.

The board also has a digital ouput port at address 18h, this is used to control the loading motor.

CALL MOTOR(N)

N=0 motor stop
N=1 motor pull
N=2 motor push
N=3 motor stop

MICRO INTERFACE ROUTINES FOR ADC AND DAC

```
        program testda
        ichan=1
        do 100 i=1,512
        volts=(i-257)/100.
        write(1,2)volts
2       format('+',f10.5)
        call dac(ichan,volts)
        do 100 j=1,10000
100     continue
        call exit
        end
        SUBROUTINE GAINAD(IG,IGADC,IGA)
        INTEGER*2 IGS(16)
        BYTE IGA
        DATA IGS/1,2,4,4,4,8,16,16,32,64,128,128,256,512,1024,1024/
        DO 1 I=1,16
        IF(IG.LT.IGS(I)) GOTO 100
1       CONTINUE
        IGA=14
        IGADC=1024
        RETURN
100     IGA=I-2
        IGADC=IGS(I-1)
        RETURN
        END
        SUBROUTINE DAC(ICHAN,VOLTS)
C       Routine to write to a Cromemco D+7A DAC
C
C       ICHAN is the channel number (1 to 7) (I*2)
C       VOLTS is the output voltage +/- 2.56V (R*4)
C       As the voltage cannot necessarily be set
C       absolutely, the actual voltage is returned.
C
        BYTE CH,IV
        IF(ICHAN.LT.1.OR.ICHAN.GT.7) GOTO 101
        IF(VOLTS.GE.2.54.OR.VOLTS.LT.-2.56) GOTO 100
        vadd=0.5
        if(volts.le.-0.01)vadd=-0.5
        IV=(VOLTS/.02)+vadd
        VOLTS=.02*IV
        CH=ICHAN+24
        CALL OUT(CH,IV)
        CALL OUT(CH,IV)
        RETURN
100     WRITE(1,1) VOLTS,CHAN
1       FORMAT(' DAC voltage ',f10.4,' out of range on channel',I5)
        RETURN
101     WRITE(1,3) ICHAN
3       FORMAT(' Channel',i5,' doesn''t exist')
        RETURN
        END
c
c
c
        SUBROUTINE MOTOR(I)
```

```
C        ROUTINE TO CONTROL MOTOR
C        0 & 3 SWITCH OFF
C        1 APPLIES LOAD
C        2 REDUCES LOAD
         BYTE I
         CALL OUT(24,I)
         RETURN
         END
C
C
C
```

```
                .z80
; PROGRAM TO READ CHANNEL OF ADC
; CALLED FROM FORTRAN
; CALL ADREAD(ICHAN,IGAIN,IVALUE)
ADREAD::
                push    hl
                push    de
                push    bc
                push    bc      ;save address of IVALUE
                push    de      ;save address of IGAIN
                ld      a,(hl)  ;load a reg. with ICHAN
                out     (0a1h),a        ;output channel to ADC
                pop     hl      ;recall address IGAIN
                ld      a,(hl)  ;load a reg. with IGAIN
                add     a,64    ;set operation mode bit
                out     (0a0h),a        ;set gain and mode ADC
                ld      hl,0
                ld      b,0
        loop:   ld      a,0
                out     (0a2h),a        ;make a to d conversion
                in      a,(0a2h)        ;read low byte from ADC
                ld      e,a
                in      a,(0a3h)        ;read high byte from ADC
                ld      d,a
                add     hl,de
                inc     b
                ld      a,b
                cp      8
                jp      m,loop
                push    hl
                pop     de
                pop     hl
                ld      (hl),e
                inc     hl
                ld      (hl),d
                pop     bc
                pop     de
                pop     hl
                ret
        wait::
                push    hl
                ld      a,(hl)
                ld      b,a
        delay:  ld      de,-1
        lop1:   ld      hl,1739
        lop2:   add     hl,de
                jr      c,lop2
                djnz    lop1
                pop     hl
                ret
                end
```

Appendix 5 - Computer Specification.

Appendix 5.

Computer Specification.

TRANSAM Tuscan S100 bus Z80A processor (4MHz) with two 5" Double sided/Double Density Floppy disks each holding 395k and 60k of RAM Operating under CP/M version 2.0

Additional Equipment.

California Data Company 12-bit programmable gain Analogue to Digital converter (16 Channel Multiplexed).

Matrox ALT-512 Graphics board with 512x256 dot resolution.

Transam VB4 VDU board

(Transam IO-3 8-bit ADC/DAC, 8 parallel ports, sound generator, system clock and timers)

EPSON MX-100 Dot matrix printer.

Transtec 1200 Monitor (Graphics)
BMC monitor (VDU).

Penny and Giles potentiometers, 1k 25mm stroke 6 off.
Smiths Industries 24V electric linear actuators 2 off.

Appendix 5 - Computer Specification.

Software.

CP/M Version 2.0	(Digital Research)
FORTRAN 80	(Microsoft)
MACRO 80	(Microsoft)
EDIT 80	(Microsoft)
BASIC	(Transam)

Universal graphics interpreter for Matrox ALT512 (not used).

Appendix 6 - Oxford Polytechnic beam results.

Appendix 6.

Oxford Polytechnic Beam Stiffness Results.

Beam Material	Joining Method	Bending Stiffness kN/mm	Torsional Stiffness kNm/rad.
1.2mm Aluminium	Rivets	1.6	2.6
	Spotwelds	1.9	5.3
	2 part epoxy CC	2.8	6.7
	1 part epoxy HC	3.1	11.1
	Toughened acrylic	3.7	6.3
0.9mm Mild Steel	Rivets	1.8	2.5
	Spotwelds	1.9	9.5
	2 part epoxy CC	2.9	11.6
	1 part epoxy HC	4.2	8.2
	Toughened acrylic	2.85	11.0

Notes:

Spotwelds and rivets at 25mm pitch.

CC = Cold Cured

HC = Hot Cured

Appendix 6 - Oxford Polytechnic beam results.

Mechanical properties of box sections.

1.2mm Aluminium

Joining Medium	Flexural Stiffness kN/mm	Torsional Stiffness kNm/rad.
Riveted	1.58	2.63
Spotweld	1.04	5.31
ESP 105	3.87	11.56
EC 2214	2.46	10.60
E 32	2.82	6.65
Bondall	2.55	7.24
F 241	4.80	5.44

0.9mm Aluminium

Riveted	0.7	1.79
ESP 105	2.04	5.12
EC 2214	2.42	4.63

Appendix 6 - Oxford Polytechnic beam results.

0.9mm Mild Steel

Joining Medium	Flexural Stiffness kN/mm	Torsional Stiffness kNm/rad.
Riveted	1.80	2.52
Spotweld	1.93	9.47
ESP 105	4.75	7.87
EC 2214	3.64	8.57
AV 100	4.0	9.93
F 241	2.85	11.0
E 32	1.84	13.3
1.35mm Mild Steel		
Riveted	2.63	3.21
Spotweld	4.83	10.34
ESP 105	7.24	12.26

Appendix 7.

Calculation of equivalent bars to replace a shear panel.

Consider the panel in Figure ap7.1(a) is to be replaced with a frame ap7.1(b). Using the theory of complementary work:

$$u = \int_v \int_o^{\gamma} T d\gamma \, dv$$

the shear energy for each case is as follows:

Panel	Frame
	diagonal strain= δ/L now $(DB')^2=(DC)^2+(CB')^2$ i.e. $(L+\delta)^2=(b+\delta_1)^2+a^2$ expanding: $L^2+2L+{}^2=b^2+2b\delta_1+\delta_1{}^2+a^2$ or: $L\delta=b\delta_1$ since $L^2=a^2+b^2$ and $\delta^2 \quad \delta_1{}^2=0$ thus: $\delta = b/L\delta_1$ now for small γ, $\delta_1=a\gamma$ $\therefore \delta = ab\gamma/L$ $\varepsilon = \delta/L = ab\gamma/L$ $a/L=\sin\theta$ and $b/L=\cos\theta$ .. $\varepsilon=\gamma\sin\theta\cos\theta$ where ε is strain in the bar and δ is change in length
$u = \int_v \int_o^{\gamma} T d\gamma dv$	$u = \int_v \int_o^{\varepsilon_1} \sigma d\varepsilon dv$

Appendix 7 - Replacing shear panels with bars.

$$u = \int_v \int_0^\gamma G\gamma d\gamma \, dv$$

Since $\tau = G\gamma$

where: τ = shear stress
γ = shear strain
v= volume
G= shear modulus
u= complementary energy

$$U = \int_v G \frac{\gamma^2}{2} dv = \frac{abt}{2} G\gamma^2$$

where: abt= volume
a= width
b= length
t= thickness

$$u = \int_v \int_0^\varepsilon \varepsilon E d\varepsilon \, dv$$

Since $\sigma = \varepsilon E$

where: σ = tensile stress
ε = tensile strain
v= volume
E= Youngs modulus
u=complementary energy

$$U = \int_v E \frac{\varepsilon^2}{2} dv = \frac{E}{2} AL\varepsilon^3$$

where: A= area of bar
l= length of bar

$U = 1/2 \, EAL \sin^2\theta \cos^2\theta\gamma^2$

==

Equating the two results:

$1/2 \, EAL\gamma^2 \sin^2\theta \cos^2\theta = 1/2 \, abtG\gamma^2$
this gives an equivalent area of a bar to replace a panel in shear as:

$A = abt/2(1+\nu)L\sin^2\theta\cos^2\theta = bt/2(1+\nu)\sin\theta\cos^2\theta$

For two diagonal bars, the strain energy is doubled
$A = abt/4(1+\nu)L\sin^2\theta\cos^2\theta = bt/4(1+\nu)\sin\theta\cos^2\theta$

Appendix 7 - Replacing shear panels with bars.

Example:

The roof of ECV3 has mean dimensions: length = 1.22m

width = 1.09m

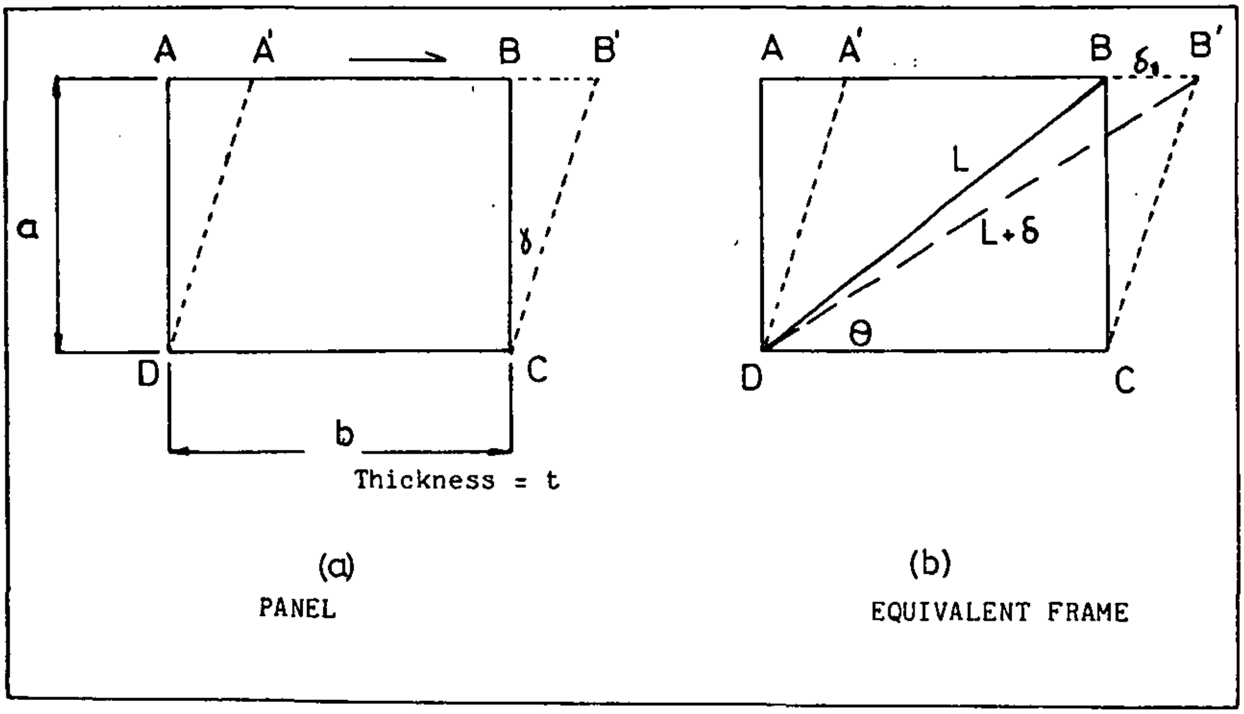
t = 0.001m

giving 0 = 41.78, and the cross sectional area for a pair of diagonal bars:

$A=1.22\mathrm{x}0.001/4(1+0.34)\sin 41.78\cos^2 41.78 = \underline{0.000614\mathrm{m}}$

Appendix 7 - Replacing shear panels with bars.

Figure ap7.1 Replacement bars for shear panel.

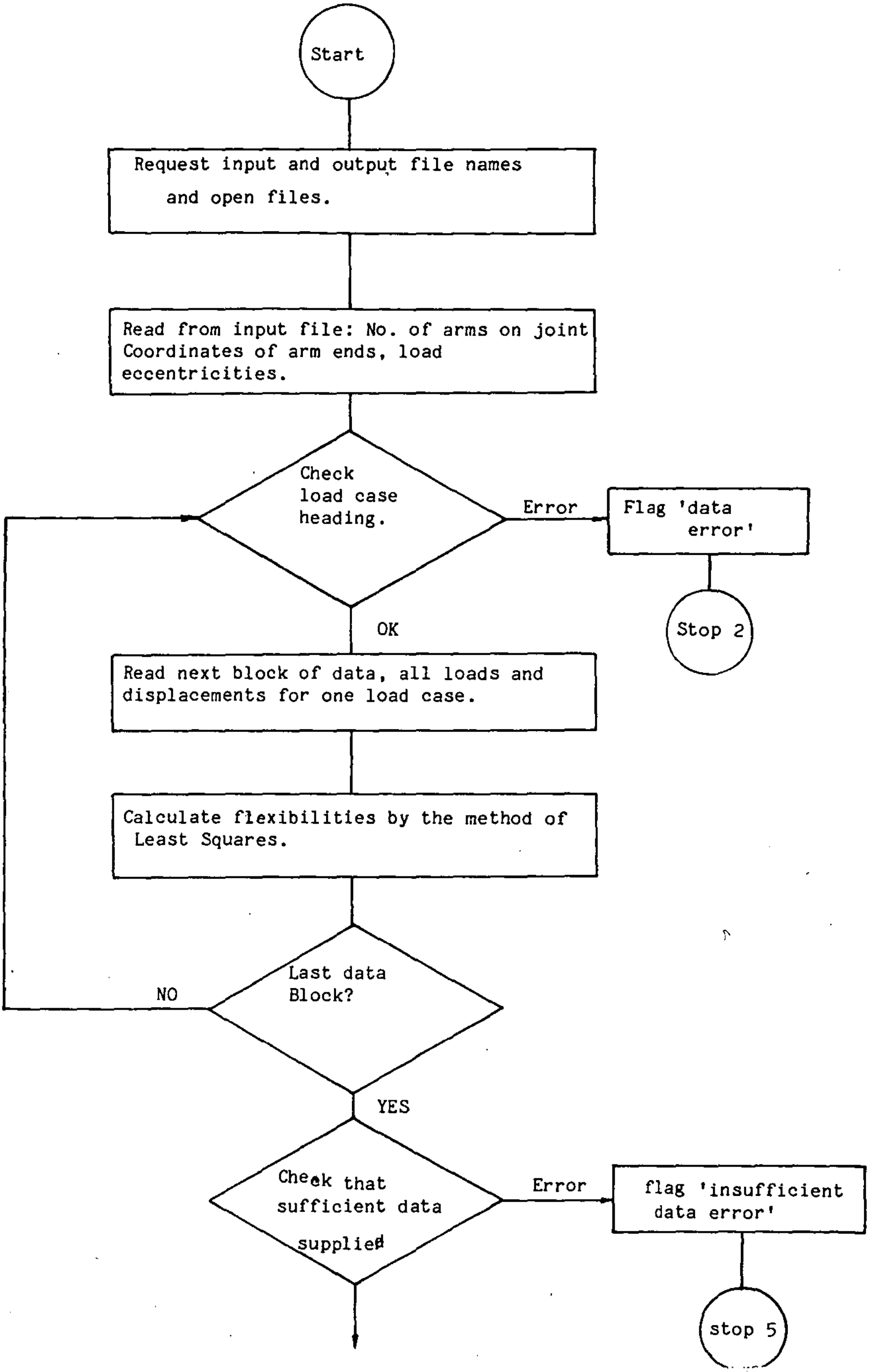


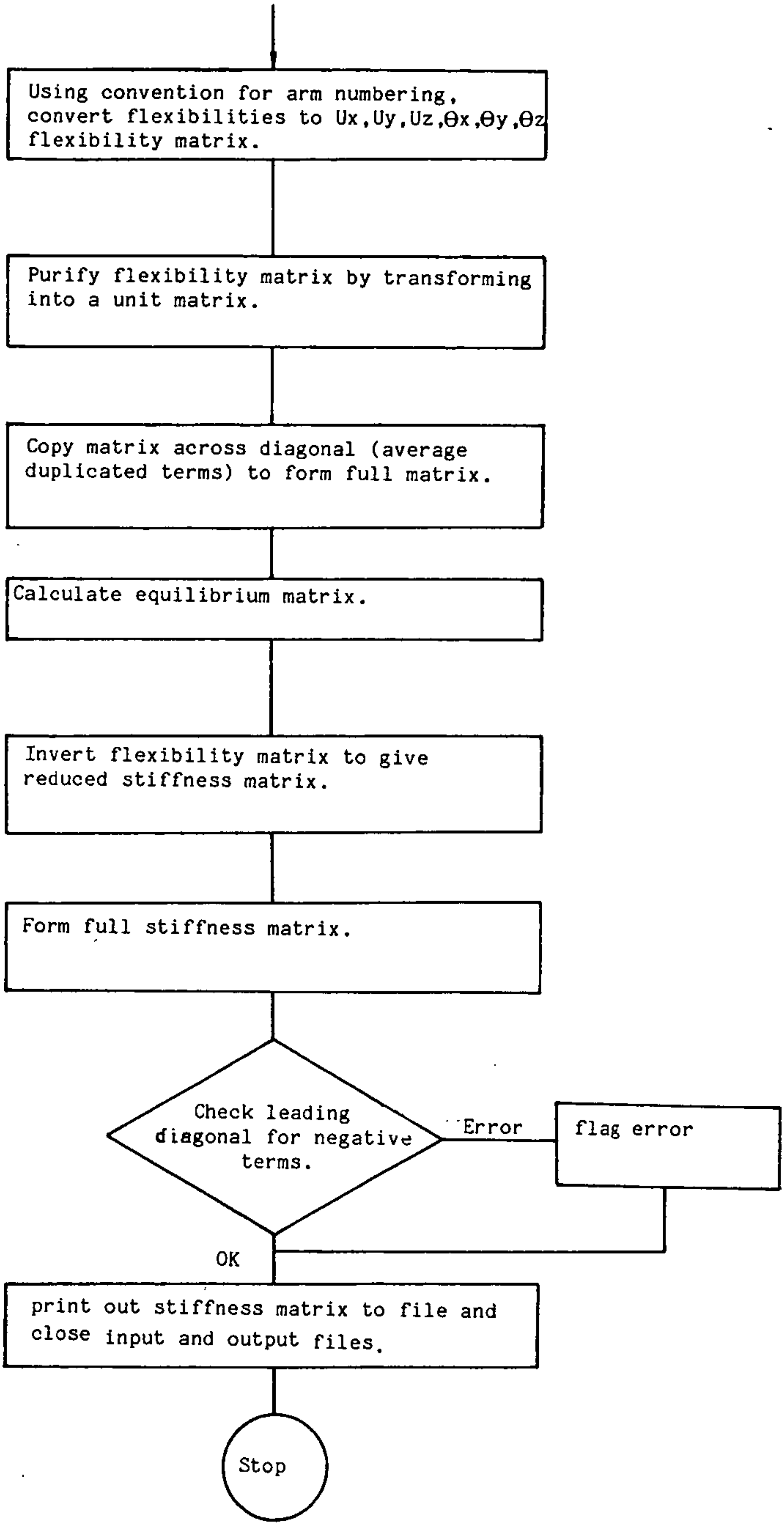
Appendix 8.

Program to generate a full stiffness matrix of a vehicle joint from experimental data.

The program was designed to take the experimental data of load v displacement and produce a full stiffness matrix. The program can analyse a 2 or 3 arm joint. A two arm joint requires a minimum of 21 measurements and a three armed joint requires 78 measurements. Each measurement must be as accurate as possible, and to ensure this the program calculates a least squares best straight line through the points from each loading cycle. If the points do not fit closely to a straight line then a warning is flagged for a correlation below 0.99 and an error below 0.95.

The program next proceeds to purify all load cases and produce a reduced flexibility matrix. This is then inverted and multiplied by the equilibrium matrix as shown in chapter 7, giving the final stiffness matrix. Here again a check is made for negative leading diagonal terms and an error is flagged.


Start
Request input and output file names and open files.
Read from input file: No. of arms on joint Coordinates of arm ends, load eccentricities.
Check load case heading.
Error
Flag 'data error'
Stop 2
OK
Read next block of data, all loads and displacements for one load case.
Calculate flexibilities by the method of Least Squares.
Last data Block?
NO
YES
Check that sufficient data supplied
Error
flag 'insufficient data error'
stop 5


Using convention for arm numbering, convert flexibilities to Ux,Uy,Uz,Θx,Θy,Θz flexibility matrix.
Purify flexibility matrix by transforming into a unit matrix.
Copy matrix across diagonal (average duplicated terms) to form full matrix.
Calculate equilibrium matrix.
Invert flexibility matrix to give reduced stiffness matrix.
Form full stiffness matrix.
Check leading diagonal for negative terms.
Error
flag error
OK
print out stiffness matrix to file and close input and output files.
Stop

```
      REAL A1(18,18),A2(18,18),A(18,18),AT(18,18),K11P(6,12)
      REAL X(3),Y(3),Z(3),H(6,12),K22(12,12),K12(6,12)
      REAL K21(12,6),K11(6,6),K(18,18),HT(12,6)
      INTEGER NPNTS,NLOAD,TYPE,CODE,IFIL(8),OFIL(8),PNTS(16)
      REAL LOAD(20),DEFL(20,16),LOADS(12),R(12,12),U(12,12)
      REAL U$(16,16)
      LOGICAL LPNTS(12),L(16,16)
      INTEGER LDPT(4),I1(16)
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
C
C OPEN INPUT AND OUTPUT FILES
C
      WRITE(1,876)
876   FORMAT('  ',///,' ******************************************
',/,
     +' **                                      **',/,
     +' **   JOINT MATRIX CALCULATING PROGRAM   **',/,
     +' **                                      **',/,
     +' ******************************************',/,
     +///)
      CALL RNAM$A('Enter data filename ',20,A$FUPP,IFIL,16)
      CALL SRCH$$(K$READ,IFIL,16,1,TYPE,CODE)
      CALL ERRPR$(K$IRTN,CODE,'Opening input file',18,0)
      CALL RNAM$A('Enter name for output file ',27,A$FUPP,OFIL,16)
      CALL SRCH$$(K$WRIT,OFIL,16,2,TYPE,CODE)
      CALL ERRPR$(K$IRTN,CODE,'Opening file for output',23,0)
      WRITE(1,765)
765   FORMAT(' ===================================================
=',/,
     +'     ****  DATA CHECHED FOR BAD CORRELATION  ****',/,
     +' ====================================================',/)
      I=0
C
C READ IN THE NUMBER OF ARMS ON THE JOINT
C
      READ(5,*)NARMS
      NAEGT=NARMS*8-8
      NASIX=NARMS*6-6
C
C READ IN COORDINATES OF ARMS ORDER: FIXED,1,2 AND LOAD ECCENTRICI
TY
C
      READ(5,*)((X(J),Y(J),Z(J)),J=1,NARMS),ECC
C
C READ CHECK STRING TO ENSURE THAT CORRECT AMOUNT OF DATA READ IN
C
200   READ(5,2000,END=1000)LDPT
2000  FORMAT(4A2)
      IF(LDPT(1).EQ.'LO')GOTO 2002
      IF(LDPT(1).EQ.'EN')GOTO 1000
      WRITE(1,2001)I,LDPT
2001  FORMAT('LOAD CASE ',I2,' ERROR',/,'SUBSEQUENT LOAD STATEMENT
 '
```

```
     +'READ AS ',4A2)
      STOP 2
C
C READ IN LOAD AND DISPLACEMENT DATA
C
2002  READ(5,*,END=1000)(LOADS(J),J=1,NASIX),NLOAD,(LOAD(J),J=1,NL
OAD),N
     +PNTS,(PNTS(J),J=1,NPNTS),((DEFL(J,K$),K$=1,NPNTS),J=1,NLOAD)
C
C   I    IS THE LOAD CASE COUNTER
C
      I=I+1
C
C SET UP ARRAY CONTAINING COLUMN INFO FOR DISPLACEMENTS
C
      DO 12 J=1,NPNTS
12    L(I,PNTS(J))=.TRUE.
      NPNT$=NPNTS
5     IF(LPNTS(NPNT$))NPNT$=NPNT$-1
      IF(LPNTS(NPNT$))GOTO 5
      LPNTS(NPNT$)=.TRUE.
      DO 7 J=1,NASIX
7     R(I,J)=LOADS(J)
      DO 9 K$=1,NPNTS
      SX=0.0
      SY=0.0
      SXY=0.0
      SXSQ=0.0
      SYSQ=0.0
      N=0
      DO 9 J=1,NLOAD
      UX=LOAD(J)*9.81
      UY=DEFL(J,K$)/100000.
C
C CALCULATE BY LEAST SQUARES, THE UNIT LOAD DEFLECTION
C
      CALL LSTSQ(UX,UY,SX,SY,SXY,SXSQ,SYSQ,N,NLOAD,U$(I,K$),I,J,K$
)
9     CONTINUE
      GOTO 200
1000  CONTINUE
      CALL TNOU(' UNIT LOAD DISPLACEMENTS',24)
      WRITE(1,567) U$
567   FORMAT(' ',4E15.5)
      STOP
C
C CHECK THAT SUFFICIENT DATA HAS BEEN SUPPLIED FOR DISPLACEMENTS
C
      DO 4 I=1,12
      IF(.NOT.LPNTS(I))GOTO 1002
      GOTO 4
1002  CALL TNOU('INSUFFICIENT DATA SUPPLIED ',27)
      STOP 5
4     CONTINUE
```

```
C
C CONVERT DISPLACEMENT DATA INTO [U] MATRIX
C
1001  DO 14 I=1,NASIX
      I2=1
      DO 11 J=1,NAEGT
      IF(L(I,J))I1(J)=I2
11    IF(L(I,J))I2=I2+1
      IF(L(I,3).AND.L(I,4)) U(I,1)=(U$(I,I1(3))+U$(I,I1(4)))/2.
      IF(L(I,2).AND.L(I,1).AND.L(I,5)) U(I,2)=(U$(I,I1(2))-U$(I,I1
(1))+
     +2.*U$(I,I1(5)))/2.
      IF(L(I,1).AND.L(I,2).AND.L(I,6)) U(I,2)=(U$(I,I1(1))-U$(I,I1
(2))+
     +2.*U$(I,I1(6)))/2.
      IF(L(I,5).AND.L(I,6)) U(I,2)=(U$(I,I1(5))+U$(I,I1(6)))/2.
      IF(L(I,1).AND.L(I,2)) U(I,3)=-(U$(I,I1(1))+U$(I,I1(2)))/2.
      IF(L(I,1).AND.L(I,2)) U(I,4)=(U$(I,I1(1))-U$(I,I1(2)))/(2.*E
CC)
      IF(L(I,7).AND.L(I,3).AND.L(I,4)) U(I,5)=-(2*U$(I,I1(7))-U$(I
,I1(3)
     +)-U$(I,I1(4)))/(2.*ECC)
      IF(L(I,8).AND.L(I,3).AND.L(I,4)) U(I,5)=-(U$(I,I1(3))+U$(I,I
1(4))
     +-2*U$(I,I1(8)))/(2.*ECC)
      IF(L(I,7).AND.L(I,8)) U(I,5)=(U$(I,I1(8))-U$(I,I1(7)))/(2.*E
CC)
      IF(L(I,3).AND.L(I,4)) U(I,6)=(U$(I,I1(3))-U$(I,I1(4)))/(2.*E
CC)
      IF(L(I,13).AND.L(I,14)) U(I,7)=-(U$(I,I1(13))+U$(I,I1(14)))/
2.
      IF(L(I,10).AND.L(I,9).AND.L(I,13)) U(I,7)=-(U$(I,I1(10))-U$(
I,I1(9
     +))+2.*U$(I,I1(13)))/2.
      IF(L(I,9).AND.L(I,10).AND.L(I,14)) U(I,7)=(U$(I,I1(9))-U$(I,
I1(10
     +))-2*U$(I,I1(14)))/2.
      IF(L(I,11).AND.L(I,12)) U(I,8)=(U$(I,I1(11))+U$(I,I1(12)))/2
.
      IF(L(I,9).AND.L(I,10)) U(I,9)=-(U$(I,I1(9))+U$(I,I1(10)))/2.
      IF(L(I,11).AND.L(I,12).AND.L(I,15)) U(I,10)=(-U$(I,I1(11))-U
$(I,I1
     +(12))+2*U$(I,I1(15)))/(2.*ECC)
      IF(L(I,11).AND.L(I,12).AND.L(I,16)) U(I,10)=-(-U$(I,I1(11))-
U$(I,I
     +1(12))+2*U$(I,I1(16)))/(2.*ECC)
      IF(L(I,15).AND.L(I,16)) U(I,10)=(U$(I,I1(15))-U$(I,I1(16)))/
(2.*EC
     +C)
      IF(L(I,9).AND.L(I,10)) U(I,11)=(U$(I,I1(9))-U$(I,I1(10)))/(2
.*ECC)
      IF(L(I,11).AND.L(I,12)) U(I,12)=(U$(I,I1(11))-U$(I,I1(12)))/
(2.*EC
     +C)
```

```
14    CONTINUE
10000 FORMAT(2(5X,E11.4))
C
C SEPERATE LOAD CASES
C
      CALL PURER(R,U,NASIX)
C
C  FORM DIAGONAL LOAD MATRIX
C
      CALL DIAGR(R,U,NASIX)
C
C FORM FULL DISPLACEMENT MATRICES
C
      DO 1 J=1,NASIX
      DO 1 K$=1,NASIX
      IF(U(K$,J).EQ.0.0)U(K$,J)=U(J,K$)
1     CONTINUE
      CALL TNOU(' NON-AVERAGED FLEXIBILITY MATRIX',32)
      WRITE(1,777) U
C
C   AVERAGE ACROSS DIAGONAL OF U
C
      DO 80 I=1,NASIX
      DO 80 J=1,NASIX
      U(I,J)=(U(I,J)+U(J,I))/2.
      U(J,I)=U(I,J)
80    CONTINUE
C
C START OF SOLUTION TO FIND [K]
C
      WRITE(1,764)
764   FORMAT(' ===================================================
==')
      CALL TNOU(' ++++ FLEXIBILITY MATRIX ++++',29)
      DO 776 I=1,12
776   WRITE(1,777) (U(I,J),J=1,I)
777   FORMAT(6E11.4)
      N=NARMS*6
      N2=NASIX
      N1=N-N2
      CALL HMAT(X,Y,Z,H)       /* CALCULATE EQUILIBRIUM MATRIX H
C       CALL LJPMI(U,N2)         /* INVERT DISPLACEMENT MATRIX
C       CALL LJPMM(R,U,K22,N2,N2,N2)     /*CALCULATE [K22]=[R]*[U]
-1
      CALL MATDIV(Z,R,U,NASIX,NASIX)
      DO 92 I=1,NASIX
      DO 92 J=1,NASIX
      K22(I,J)=R(I,J)
92    CONTINUE
      CALL LJPMT(H,HT,N1,N2)    /* CALCULATE TRANSPOSE OF [H]=[HT]
      CALL LJPMM(H,K22,K12,N1,N2,N2)
      CALL LJPMN(K12,N1,N2)            /* CALCULATE [K12]=-[H]*[K22
]
      CALL LJPMM(H,K22,K11P,N1,N2,N2)
      CALL LJPMM(K11P,HT,K11,N1,N2,N1)     /* CALCULATE [K11]=[H]*
[K22]*[HT]
      CALL LJPMA(K11,K12,K22,K,N1,N2,N)   /* FORM [K]
      WRITE(6,103)
103   FORMAT('[K]')
C
C   CHECK FOR NEGATIVE LEADING DIAGONAL TERMS
C
      WRITE(1,764)
      WRITE(1,763)
763   FORMAT(' **  K CHECKED FOR NEGATIVE LEADING DIAGONAL TERMS *
*',/,
     +' ====================================================')
      DO 33 I2=1,N
33    IF(K(I2,I2).LT.0.0) WRITE(1,761) I2,K(I2,I2)
761   FORMAT(' **ERROR**  Diagonal term',I3,' = ',E11.4)
      DO 101 I=1,N
101   WRITE(6,100)I,(K(I,J),J=1,I)
100   FORMAT(I3,' : ',6(1X,G11.4))
      CALL SRCH$$(K$CLOS,0,0,1,I1,I2)
      CALL SRCH$$(K$CLOS,0,0,2,I1,I2)
      WRITE(1,762)
762   FORMAT(//,10X,' ++ JOB.COMPLETED ++',/,11X,'---------------
---')
      CALL EXIT
      END
      SUBROUTINE HMAT(X,Y,Z,H)
      REAL X(3),Y(3),Z(3),H(6,12)
      DO 1 I=1,6
      J=I+6
      H(I,I)=1.0
1     H(I,J)=1.0
      H(4,2)=Z(1)-Z(2)
      H(4,3)=Y(2)-Y(1)
      H(4,8)=Z(1)-Z(3)
      H(4,9)=Y(3)-Y(1)
      H(5,1)=Z(2)-Z(1)
      H(5,3)=X(1)-X(2)
      H(5,7)=Z(3)-Z(1)
      H(5,9)=X(1)-X(3)
      H(6,1)=Y(1)-Y(2)
      H(6,2)=X(2)-X(1)
      H(6,7)=Y(1)-Y(3)
      H(6,8)=X(3)-X(1)
      RETURN
      END
      SUBROUTINE LJPMT(A,AT,N1,N2)
      REAL A(N1,N2),AT(N2,N1)
      DO 1 I=1,N1
      DO 1 J=1,N2
1     AT(J,I)=A(I,J)
      RETURN
      END
      SUBROUTINE LJPMM(A,A1,A2,N1,N2,N3)
      REAL A(N1,N2),A1(N2,N3),A2(N1,N3)
      DO 1 I=1,N1
      DO 1 J=1,N2
      A2(I,J)=0.0
      DO 1 K=1,N2
1     A2(I,J)=A2(I,J)+A(I,K)*A1(K,J)
      RETURN
      END
      SUBROUTINE LJPMN(A,N1,N2)
      REAL A(N1,N2)
      DO 1 I=1,N1
      DO 1 J=1,N2
1     A(I,J)=-A(I,J)
      RETURN
      END
      SUBROUTINE LJPMA(K11,K12,K22,K,N1,N2,N)
      REAL K11(N1,N1),K12(N1,N2),K22(N2,N2)
      REAL K(N,N)
      DO 1 I=1,N1
      DO 1 J=1,N1
1     K(I,J)=K11(I,J)
      DO 2 I=1,N2
      DO 2 J=1,N2
      N3=N1+I
      N4=N1+J
2     K(N3,N4)=K22(I,J)
      DO 3 I=1,N1
      DO 3 J=1,N2
      N6=J+N1
      K(I,N6)=K12(I,J)
      K(N6,I)=K(I,N6)
3     CONTINUE
      RETURN
      END
      SUBROUTINE LSTSQ(UX,UY,SX,SY,SXY,SXSQ,SYSQ,N,N1,DFLTN,I,J,K$
)
      SX=SX+UX
      SY=SY+UY
      SXY=SXY+UX*UY
      SXSQ=SXSQ+UX**2
      SYSQ=SYSQ+UY**2
      N=N+1
      IF(N.NE.N1)RETURN
      DFLTN=(SXY-(SX*SY)/N)/(SXSQ-(SX**2/N))
      SDX=SQRT((SXSQ-SX**2/N)/(N-1))
      SDY=SQRT((SYSQ-SY**2/N)/(N-1))
      IF(SDY.EQ.0.0) COR=1
      IF(SDY.EQ.0.0) RETURN
      COR=DFLTN*SDX/SDY
      IF(ABS(COR).LT.0.95) CALL TNOUA(' **ERROR** ',11)
      IF(ABS(COR).GE.0.95.AND.ABS(COR).LT.0.99) CALL TNOUA(' *WARN
ING* '
     +,11)
      IF(ABS(COR).LT.0.99)WRITE(1,99) I,K$,COR
```

```
99     FORMAT(' Load case ',I3,'  Point pos. ',I3,'   Correlation='
,F6.4)
       RETURN
       END
       SUBROUTINE PURER(R,U,NASIX)
       DIMENSION R(NASIX,NASIX),U(NASIX,NASIX),R$(12,12),U$(12,12)
       INTEGER NASIX
       LOGICAL C(12,12),L(12),C1(12),C2(12,12)
       DO 1 I=1,NASIX
       DO 1 J=1,NASIX
1      IF(R(I,J).NE.0.0) C(I,J)=.TRUE.
C
C
C SEARCH ALONG LINES TO FIND BOTH LOADS
C
C
       I=0
4      J1=0
       I=I+1
       IF(I.GT.NASIX)GOTO 30
       DO 22 J=1,NASIX
       IF(C(I,J))GOTO 3
       GOTO 22
3      IF(J1.EQ.0)J1=J
       J2=J
22     CONTINUE
       I2=0
       I11=I+1
       DO 5 I1=I11,NASIX
       IF(L(I1))GOTO 5
       IF(C(I1,J1).AND.C(I1,J2)) I2=I1
5      CONTINUE
       IF(I2.EQ.0) GOTO 4
       DO 20 I1=1,NASIX
       R$(I,I1)=R(I,I1)+R(I2,I1)
C       IF(U(I,I1).EQ.0.0.OR.U(I2,I1).EQ.0.0) U$(I,I1)=0.0
C       IF(U(I,I1).NE.0.0.AND.U(I2,I1).NE.0.0) U$(I,I1)=U(I,I1)+U(I
2,I1)
       U$(I,I1)=U(I,I1)+U(I2,I1)
       R$(I2,I1)=R(I,I1)-R(I2,I1)
C       IF(U(I,I1).EQ.0.0.OR.U(I2,I1).EQ.0.0) U$(I2,I1)=0.0
C       IF(U(I,I1).NE.0.0.AND.U(I2,I1).NE.0.0) U$(I2,I1)=U(I,I1)-U(
I2,I1)
20      U$(I2,I1)=U(I,I1)-U(I2,I1)
       L(I)=.TRUE.
       L(I2)=.TRUE.
       GOTO 4
30     DO 40 I=1,NASIX
       IF(.NOT.L(I)) GOTO 40
       DO 40 J=1,NASIX
       IF(ABS(R$(I,J)).GT.1E-6) C2(I,J)=.TRUE.
       IF(C(I,J)) C1(J)=.TRUE.
40     CONTINUE
       DO 50 J=1,NASIX
```

```
      IF(C1(J)) GOTO 50
      DO 50 I=1,NASIX
      IF(.NOT.C(I,J)) GOTO 50
      IF(L(I)) GOTO 50
      DO 50 J1=1,NASIX
      IF(J1.EQ.J) GOTO 50
      IF(.NOT.C(I,J1)) GOTO 50
      DO 50 I1=1,NASIX
      IF(.NOT.L(I1)) GOTO 50
      IF(.NOT.C2(I1,J1)) GOTO 50
      IF(C1(J))GOTO 50
      A=R$(I1,J1)/R(I,J1)
      DO 47 J2=1,NASIX
      R$(I,J2)=A*R(I,J2)-R$(I1,J2)
C      IF(U(I,J2).EQ.0.0.OR.U(I1,J2).EQ.0.0) U$(I,J2)=0.0
C      IF(U(I,J2).NE.0.0.AND.U(I1,J2).NE.0.0)U$(I,J2)=A*U(I,J2)-U$
(I1,J2)
47    U$(I,J2)=A*U(I,J2)-U$(I1,J2)
      L(I)=.TRUE.
50    CONTINUE
      DO 60 I=1,NASIX
      DO 60 J=1,NASIX
      R(I,J)=R$(I,J)
      U(I,J)=U$(I,J)
60      CONTINUE
      RETURN
      END
      SUBROUTINE DIAGR(R,U,NASIX)
      DIMENSION R(NASIX,NASIX),U(NASIX,NASIX),R$(12,12),U$(12,12)
      L1=1
8     J1=0
      IF(L1.GT.NASIX)GOTO 1
2     J1=J1+1
      IF(J1.GT.NASIX)GOTO 6
      IF(ABS(R(J1,L1)).GT.1E-6) GOTO 3
      GOTO 2
3     DO 10 L2=1,NASIX
      R$(L1,L2)=R(J1,L2)
10    U$(L1,L2)=U(J1,L2)
      L1=L1+1
      GOTO 8
6     WRITE(1,11)L1
11    FORMAT('NO LOAD CASE FOR COLUMN ',I2)
C
      STOP 3
1     DO 4 I=1,NASIX
      DO 4 J=1,NASIX
      U(I,J)=U$(I,J)/R$(I,I)
      R(I,J)=0.0
      R(I,I)=1.0
4     CONTINUE
      RETURN
      END
      SUBROUTINE MATINV(DET,A,B,N)
      DIMENSION     A(N,N),     B(N,N)
      CALL UNIT(A,N)
      CALL MATDIV(DET,A,B,N,N)
      RETURN
      END
      SUBROUTINE MATMUL(A,B,C,N,K,M)
C-----A(N BY M)=B(N BY K)*C(K BY M)
      DIMENSION A(1),B(1),C(1)
      IC=1
      IA=1
         DO 130 L1=1,M
            DO 120 L2=1,N
            IB=L2
            X=0.0
               DO 110 L3=1,K
               X=X+B(IB)*C(IC)
               IB=IB+N
               IC=IC+1
  110       CONTINUE
            A(IA)=X
            IA=IA+1
            IC=IC-K
  120    CONTINUE
         IC=IC+K
  130 CONTINUE
      RETURN
      END
      SUBROUTINE TRAMUL(A,B,C,N,K,M)
      DIMENSION A(1),B(1),C(1)
C-----A(N BY K)=B(N BY M)*TRANSPOSE OF C(K BY M)
      IA=1
         DO 130 L1=1,K
            DO 120 L2=1,N
            IC=L1
            IB=L2
            X=0.0
               DO 110 L3=1,M
               X=X+B(IB)*C(IC)
               IB=IB+N
               IC=IC+K
  110          CONTINUE
            A(IA)=X
            IA=IA+1
  120       CONTINUE
  130    CONTINUE
      RETURN
      END
      SUBROUTINE MATDIV(Z,A,B,M,N)
      DIMENSION     A(M,N),     B(M,M)
      IF(M.EQ.1) GOTO 230
      Z=1.0
      M1=M-1
         DO 180 L1=1,M1
         X=0
```

```
          DO 110 L2=L1,M
          IF(DABS(B(L2,L1)).LT.DABS(X)) GOTO 110
          K=L2
          X=B(L2,L1)
110       CONTINUE
       Z=Z*X
       IF(K.EQ.L1) GOTO 120
       Z=-Z
120    X=1.0/X
          DO 130 L2=L1,M
          Y=X*B(K,L2)
          B(K,L2)=B(L1,L2)
          B(L1,L2)=Y
130       CONTINUE
          DO 140 L2=1,N
          Y=A(K,L2)*X
          A(K,L2)=A(L1,L2)
          A(L1,L2)=Y
140       CONTINUE
       I1=L1+1
          DO 170 L2=I1,M
          Y=B(L2,L1)
             DO 150 L3=I1,M
             B(L2,L3)=B(L2,L3)-Y*B(L1,L3)
150          CONTINUE
             DO 160 L3=1,N
             A(L2,L3)=A(L2,L3)-Y*A(L1,L3)
160          CONTINUE
170       CONTINUE
180    CONTINUE
    Z=Z*B(M,M)
    X=1.0/(B(M,M))
       DO 190 L1=1,N
190    A(M,L1)=A(M,L1)*X
       DO 220 L1=2,M
       KD=M-L1+2
       KD1=KD-1
          DO 210 L2=1,KD1
          X=B(L2,KD)
             DO 200 L3=1,N
             A(L2,L3)=A(L2,L3)-A(KD,L3)*X
200          CONTINUE
210       CONTINUE
220    CONTINUE
    GOTO 250
230 Z=B(1,1)
    IF(Z.EQ.0.0) GOTO 250
       DO 240 L1=1,N
240    A(1,L1)=A(1,L1)/Z
250 RETURN
    END
    SUBROUTINE NULL(A,M,N)
    DIMENSION A(1)
    MN=M*N
```

```
      DO 110 L1=1,MN
      A(L1)=0.0
110   CONTINUE
   RETURN
   END
   SUBROUTINE UNIT(A,N)
   DIMENSION A(1)
   CALL NULL(A,N,N)
   IA=1
      DO 110 L1=1,N
      A(IA)=1.0
      IA=IA+N+1
110   CONTINUE
   RETURN
   END
```

Appendix 9 - Joint Stiffness measurement program

Computer program to run Experimental analysis of the stiffness of a joint.

```
c       Program to run structures lab. experiment
c       for the measurement of joint stiffnesses.
c
c
c
        real*4 grad(6),cor(6),diffv(6),avg(6),avgl(6)
        real*4 dacvl(6),dacv2(6),displ(50,6),load(50,6)
        byte igadc(6),lacpt,lgraf
        integer*2 ivall(6),ival2(6),igain(6),ival(6),iv(6)
        call matrox
        write(1,99)
99      format(' ************************************',/,
     +' Program to run structural stiffness',/,
     +' experiment on a vehicle joint.',/,
     +' ************************************')
        lgraf=.false.
        call zload(lddat)
        write(1,111)
111     format('0',/,'+Enter Maximum load:')
        read(1,112) amaxld
112     format(g10.4)
        nl=0
        n2=0
        call chkok(0.0,0.0,0,0,0.0,0.0,0.0,0,nl,n2)
c
c       Set dac voltages to put lvolt across pots.
c       and read pot. voltage to find zero datum.
        do 1 i=1,6
        avg(i)=0.0
        vl=0.0
        v2=2.0
        call disp(i,vl,v2,4,ivall(i),nl)
1       continue
c       All zero datums taken and held in array ival
c
c       Apply full load and take pot. readings.
        write(1,108) lddat
108     format(' Datum load value=',i6)
        call setld(amaxld,lddat)
        write(1,109) amaxld
109     format(' Maximum load value=',f6.2)
        do 2 i=1,6
        call disp(i,vl,v2,4,ival2(i),nl)
2       continue
c
c       calculate actual voltages read
        do 3 i=1,6
        dacvl(i)=-((ivall(i)/819.2)-vl)
        dacv2(i)=dacvl(i)+(v2-vl)
        idiff=iabs(ival2(i)-ivall(i))
        if(idiff.lt.16) idiff=16
        ign=8192/idiff
        call gainad(ign,igain(i),igadc(i))
3       continue
        write(1,100) ivall,ival2,igain,igadc,dacvl,dacv2
```

```
100     format('0                   Channel',/,
    +'                      1      2      3      4      5      6    Load',/,
    +'0Gain Zero         ',6i6,/,
    +' Gain maximum      ',6i6,/,
    +' Gain chosen       ',6i6,/,
    +' Gain for ADC      ',6i6,/,
    +' DAC1 voltage set',6f6.2,/,
    +' DAC2 voltage set',6f6.2)
c
c       Set new datum values at zero load
c       with dac values set
        k=0
6       datld=0.0
        call setld(datld,lddat)
        if(lgraf) goto 17
        write(1,110) datld
110     format(' Datum zero load=',f6.2)
c       Load is zeroed
17      do 7 i=1,6
        j=0
14      call disp(i,dacv1(i),dacv2(i),igadc(i),ival1(i),n1)
        if(ival1(i).lt.1024.and.ival1(i).gt.-1024) goto 13
        if(ival1(i).gt.1024) dacv1(i)=dacv1(i)-0.02
        if(ival1(i).lt.-1024)dacv1(i)=dacv1(i)+0.02
        j=j+1
        if(j.lt.5) goto 14
        if(ival1(i).gt.1024) dacv2(i)=dacv2(i)-0.02
        if(ival1(i).lt.-1024) dacv2(i)=dacv2(i)+0.02
        if(j.lt.7) goto 14
        write(1,105) i,dacv1(i),dacv2(i),ival1(i)
105     format('0Unsuccessful at finding suitable zero voltage',/,
    +' on channel ',i2,/,
    +' Voltages ',2f7.3,'produce value ',i5,//,
    +'+Enter possible new values: ')
        read(1,106) dacv1(i),dacv2(i)
106     format(2g10.4)
        goto 14
13      diffv(i)=dacv2(i)-dacv1(i)
7       continue
        do 19 i=1,6
19      call disp(i,dacv1(i),dacv2(i),igadc(i),ival(i),n1)
        if(lgraf) goto 18
        write(1,101) dacv1,dacv2,ival
101     format(' DAC volts reset ',6f6.2,/,
    +' DAC volts 2 res ',6f6.2,/,
    +' Datum voltages  ',6i6,)
c
c       take a set of readings
c
c       increment load
18      j=0
8       reqld=(j+1)*amaxld/n2
        call setld(reqld,lddat)
c       read potentiometers
        j=j+1
        do 10 i=1,6
        call disp(i,dacv1(i),dacv2(i),igadc(i),iv(i),n1)
        call adread(7,0,ild)
102     format(' Reading ',i8,7i6)
        if(iv(i).eq.2047.or.iv(i).eq.-2048) write(1,115) i
115     format(' Over/Underflow on adc channel',i5)
        displ(j,i)=(.74447/abs(diffv(i))/igain(i))*(iv(i)-ival(i))
        load(j,i)=(ild-lddat)/38.227
10      continue
        if(lgraf) goto 16
        write(1,102) j,iv,ild
16      if(j.lt.n2) goto 8
        do 12 i=1,6
        call lstsq(load(1,i),displ(1,i),j,grad(i),cor(i),yint)
        if(lgraf) goto 12
        write(1,107) i,grad(i),cor(i)
107     format(' Gauge',i2,'  Gradient ',g10.4,' correlation ',f6.3,/)
        call chkok(load(1,i),displ(1,i),j,i,
     +grad(i),cor(i),yint,lacpt,n1,n2)
        if(.not.lacpt) goto 6
12      continue
        k=k+1
        lacpt=.true.
        do 15 i=1,6
        a=k
        avg1(i)=avg(i)
        avg(i)=(((a-1.0)*avg(i))+grad(i))/a
        if(abs(avg(i)).lt.abs(avg1(i)*.995).or.abs(avg(i)).gt.
     +abs(avg1(i)*1.005)) lacpt=.false.
15      continue
        write(1,113) avg
113     format(' Gradients= ',6g10.3)
        lgraf=.true.
        if(.not.lacpt) goto 6
        call setld(datld,lddat)
        call exit
        end
c
c
c
        subroutine disp(i,v1,v2,ign,ival,ird)
        byte ign,iadc
        vi=0.0
        iadc=i-1
        do 1 k=1,ird
        call dac(1,v1)
        call dac(2,v2)
        call adread(iadc,ign,ival)
        vi=vi+ival/8
1       continue
        ival=vi/k
        return
        end
c
c
```

```
c
          subroutine lstsq(x,y,n,grad,cor,yint)
          dimension x(n),y(n)
          sx=0.0
          sy=0.0
          sxy=0.0
          sxsq=0.0
          sysq=0.0
          do 1 i=1,n
          sx=sx+x(i)
          sy=sy+y(i)
          sxy=sxy+x(i)*y(i)
          sxsq=sxsq+x(i)**2
          sysq=sysq+y(i)**2
1         continue
          grad=(sxy-(sx*sy)/n)/(sxsq-(sx**2/n))
          yint=(sy-grad*sx)/n
          sdx=((sxsq-sx**2/n)/(n-1))
          if(sdx.lt.0.0) goto5
          sdx=sqrt(sdx)
5         sdy=((sysq-sy**2/n)/(n-1))
          if(sdy.lt.0.0) goto 6
          sdy=sqrt(sdy)
          if(sdy.le.0.0) cor=1.0
6         if(sdy.le.0.0) return
          cor=grad*sdx/sdy
          return
          end
c
c
c
          subroutine grafs(x,y,n,b,c)
          dimension x(n),y(n),x0(2),y0(2)
          xmax=x(1)
          xmin=x(1)
          ymax=y(1)
          ymin=y(1)
          do 1 i=2,n
          if(x(i).gt.xmax) xmax=x(i)
          if(x(i).lt.xmin) xmin=x(i)
          if(y(i).gt.ymax) ymax=y(i)
          if(y(i).lt.ymin) ymin=y(i)
1         continue
          ys1=xmin*b+c
          ys2=xmax*b+c
          x0(1)=xmin
          x0(2)=xmax
          y0(1)=ys1
          y0(2)=ys2
          if(ys1.gt.ymax) x0(1)=(ymax-c)/b
          if(ys1.gt.ymax) y0(1)=ymax
          if(ys1.lt.ymin) x0(1)=(ymin-c)/b
          if(ys1.lt.ymin) y0(1)=ymin
          if(ys2.gt.ymax) x0(2)=(ymax-c)/b
          if(ys2.gt.ymax) y0(2)=ymax
```

```
      if(ys2.lt.ymin) x0(2)=(ymin-c)/b
      if(ys2.lt.ymin) y0(2)=ymin
      call piccle
      call axis(170.0,70.0)
      call scal(10,10)
      call axisca(xmin,xmax,ymin,ymax)
      call grasym(x,y,n,'*')
      call gralin(x0,y0,2)
      return
      end
c
c
c
      subroutine chkok(x,y,n,m,a,b,c,l,n1,n2)
      byte l,itit(8)
      if(n1.eq.0) goto 1
      call grafs(x,y,n,a,c)
      call movto2(150.0,0.0)
      call chaarr('Load (N)',8)
      call movto2(25.0,82.0)
      call chaarr('Deflection (mm)',15)
      call movto2(100.0,86.0)
      call chaarr('Graph for potentiometer ',24)
      encode(itit,1006) m
1006  format(i2)
      call chaarr(itit,2)
      call linto2(100.0,86.0)
      call movto2(100.0,82.0)
      call chaarr('Correlation=',12)
      encode(itit,1007) b
1007  format(f6.3)
      call chaarr(itit,6)
      call movto2(100.0,78.0)
      call chaarr('Gradient=',9)
      encode(itit,1008) a
1008  format(g8.2)
      call chaarr(itit,8)
      write(1,1000)
1000  format(' ',/,'+Print graph?')
      read(1,1001) ia
1001  format(a1)
      if(ia.eq.'y') call grdump
      write(1,1002)
1002  format('+Is line acceptable? ')
      read(1,1001) ia
      l=.false.
      if(ia.eq.'y') l=.true.
      if(l) return
1     write(1,1003)
1003  format('+Number of readings? ')
      read(1,1004) n2
1004  format(i6)
      write(1,1005)
1005  format('+Number to average? ')
      read(1,1004) n1
      return
      end
c
c
c
      subroutine zload(lddat)
1     call motor(2)
      ildl=ild
      call adread(7,0,ild)
      if(ild.lt.ildl) gotol
      call motor(0)
      call adread(7,0,lddat)
      return
      end
c
c
c
      subroutine setld(x,lddat)
      ld=x*38.227
      call adread(7,0,ild)
      ldinc=ld-ild+lddat
      if(ldinc.lt.0) goto 2
1     call motor(1)
      call adread(7,0,ild)
      if((ild-lddat).lt.(ld)) goto 1
      call motor(0)
      call wait(10)
      call adread(7,0,ild)
      if((ild-lddat).lt.(ld)) gotol
3     x=(ild-lddat)/38.227
      return
2     call motor(2)
      call adread(7,0,ild)
      if((ild-lddat).gt.ld) goto 2
      call motor(0)
      call wait(10)
      call adread(7,0,ild)
      if((ild-lddat).gt.ld) goto 2
      goto 3
      end
c
c
c
```

Appendix 10.

PAFEC File deleting Program.

```
C ****************************************************
C ***  PAFEC FILE DELETEING ROUTINE  Mk VIII ***
C ***       FOR  PAFEC 75  level 3.4         ***
C ****************************************************
      INTEGER*2 TYPE,CODE,JFIL(4),IFIL(2),EA,EB,IAST2(2),LFIL(16)
      LOGICAL LOG
$INSERT SYSCOM>ERRD.F
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>A$KEYS
      DATA IAST2(1)/'**'/,IAST2(2)/'**'/,JFIL(1),JFIL(2)/0,19/
C ----- READ FILE NAME FROM COMMAND LINE
      CALL RDTK$$(1,LFIL,IFIL,2,CODE)
      IF(LFIL(2).GT.0) GOTO 12
      CALL TNOU(' If you have not yet plotted your plotting files',48)
      CALL TNOU(' Press <BREAK> now as this routine will destroy',47)
      CALL TNOU(' all G<jobname> files.',22)
      WRITE(1,111)
111   FORMAT(/,/,/,/,/)
      CALL RNAM$A(' ***  Enter job name of files to be deleted  ',44,
     +A$FUPP,IFIL,4)
12    JFIL(1)=0
      JFIL(2)=19
      CALL TSRC$$(K$READ,'TT>LJP>PAFEC.DELETE',1,JFIL,TYPE,CODE)
14    READ(5,3,END=200)JFIL
3     FORMAT(8A2)
C -----  READ FILE PAFEC.DELETE AND MODIFY ****,FOR JOB NAME ---------
-----
      CALL LSTR$A(IAST2,4,JFIL,8,I2,I3)
      CALL MSUB$A(IFIL,4,1,4,JFIL,8,I2,I3)
C -----  CHECK FOR EXISTANCE OF FILE, THEN CHECK TYPE, DELETE AS APPROP
RIATE
      CALL SRCH$$(K$CLOS,JFIL,8,0,TYPE,I1)
13     CALL SRCH$$(K$DELE,JFIL,8,0,TYPE,I1)
       IF(I1.EQ.E$DNTE)GOTO 17
      GOTO 14
200   CALL SRCH$$(K$CLOS,0,0,1,TYPE,CODE)
C -------   READ NEXT JOB NAME FROM COMMAND LINE ---------------------
-
      CALL RDTK$$(1,LFIL,IFIL,2,CODE)
      IF(LFIL(2).GT.0) GOTO 12
      CALL EXIT
C -----  DELETING ROUTINE FOR #<jobname> FILES  ----------------------
-----
17    CALL SRCH$$(K$RDWR,JFIL,8,3,TYPE,I1)
      CALL SGDR$$(K$FULL,3,EA,EB,I1)
      IF(EB.LT.0)GOTO 18
      CALL SRCH$$(K$DELE+K$ISEG,3,0,0,TYPE,I1)
      GOTO 17
18    CALL SGDR$$(K$MSIZ,3,EA,0,I1)
      CALL SRCH$$(K$CLOS,JFIL,8,3,TYPE,I1)
      GOTO 13
      END
```

```
PAFRUN
DG****
O****1
O****2
O****3
O****4
O****5
O****6
O****7
O****8
O****9
O****0
BS****
CM****
JC****
MF****
DA****
ES****
****1F
****2F
****3F
****4F
****5F
****6F
****7F
****8F
****9F
****0F
FS****
B_****1F
B_****2F
B_****3F
B_****4F
B_****5F
B_****6F
B_****7F
B_****8F
B_****9F
B_****0F
G****3
G****5
G****8
G****0
#****
B****1
B****2
B****3
B****4
B****5
B****6
B****7
B****8
B****9
B****0
```

```
D1****
D2****
T$0000
T$0001
T$0002
T$0003
T$0004
C****
$PRI1
$PRI2
$PRI3
$LOOF
$PBSH
$PEL2
$PEL3
$ALIB
$BLIB
```

PAFEC Batch Plotting Routine.

```
C     ***************************************************************
*
C     BATCH PLOTTING ROUTINE FOR PAFEC 75
C     ***************************************************************
*
      INTEGER*2 IFIL(3),IUSER(16),IUS(6),IGC,IUFD(10)
      INTEGER*2 TYPE,CODE
      LOGICAL LOG,LOGY
$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>A$KEYS
      DATA IG/'G'/
      DATA IUS(1)/'M_'/
      DATA IUS(2)/'PP'/
C
C     READ RUN STREAM FROM COMMAND LINE
C
      CALL RDTK$$(1,IUSER,IST,1,CODE)
      IF(IUSER(2).EQ.0) IST='P4'
C
C ----- OBTAIN USERNUMBER ----------------------------------
      CALL TIMDAT(IUSER,16)
      DO 4 I=3,6
      J=I+10
4     IUS(I)=IUSER(J)
C
C ----- DELETE OLD FILES ----------------------------------------
      CALL SRCH$$(K$DELE,IUS,12,0,TYPE,CODE)
      CALL SRCH$$(K$DELE,'PPSUB',5,0,TYPE,CODE)
      CALL SRCH$$(K$DELE,'PPLOT',5,0,TYPE,CODE)
C
C ----- CREATE NEW FILES ---------------------------------------------
-
      CALL SRCH$$(K$WRIT+K$NSAM+K$IUFD,'PPSUB',5,6,TYPE,CODE)
      CALL SRCH$$(K$WRIT+K$NSAM+K$IUFD,'PPLOT',5,7,TYPE,CODE)
C
C ----  CREATE DUMP FILE  -------------------------------
C
      CALL SRCH$$(K$WRIT,IUS,12,8,TYPE,CODE)
      WRITE(12,1000)
1000  FORMAT('Plot not yet produced.')
      CALL SRCH$$(K$CLOS,0,0,8,TYPE,CODE)
      CALL SATR$$(K$PROT,IUS,12,:701600000,CODE)
      CALL ERRPR$(K$NRTN,CODE,0,0,0)
C
C     FIND NAME OF HOME U.F.D.
C
      CALL GPATH$(K$HOMA,0,IUFD,20,IP,CODE)
      CALL LSTR$A('>',1,IUFD,20,IPF,IPL)
      RIPF=IPF
      IF(RIPF/2.NE.IPF/2) CALL SSUB$A(IUFD,20,1,20,1,' ')
      IF(RIPF/2.NE.IPF/2) IP=IP+1
      IPL=(IPF+1)/2
      DO 5 I=1,IPL
5     IUFD(I)='  '
```

```
      IPL=20-IP
       CALL SSUB$A(IUFD,20,1,20,IPL,' ')
      WRITE(11,7) (IUFD(I),I=1,10),IUS
      WRITE(11,21)(IUFD(I),I=1,10)
7     FORMAT('COMO ',10A2,'>',6A2,)
21     FORMAT('A ',10A2)
      WRITE(1,1)
1     FORMAT(' Enter names of files to be plotted, each name followed'/
     +' by <RETURN>, a blank line signifies the end of the list.')
2     READ(1,3)IFIL
      IGC=GCHR$A(IFIL,1)
C
C    EXTRACT FIRST CHARACTER OF FILE NAME
C
      IF(IGC.EQ.' ') GOTO 100
      IF(IGC.NE.'G')CALL TNOU('Not a PAFEC 75 plot file ......',31)
      IF(IGC.NE.IG) GOTO 2
      LOG=EXST$A(IFIL,6)
      IF(LOG)GOTO 10
      GOTO 102
C
C    CHECK FOR EXISTANCE OF FILE NAME
C
3     FORMAT(3A2)
10    WRITE(11,8)IFIL
8     FORMAT('PLOT.PAFEC',/,3A2,/,'C1051N')
      IF(LOG) CALL SATR$$(K$PROT,IFIL,6,:701600000,I1)
      CALL ERRPR$(K$NRTN,I1,0,0,0)
      GOTO 2
100   WRITE(11,9)
9     FORMAT('LOGOUT')
      WRITE(10,6) IST,IST
6     FORMAT('SUBMIT PPLOT E-PP -',A2/
     +'COMO -TTY'/
     +'QUEUE BATCH -',A2/
     +'COMI -END')
      CALL SRCH$$(K$CLOS,'PPSUB',5,6,TYPE,CODE)
      CALL SRCH$$(K$CLOS,'PPLOT',5,7,TYPE,CODE)
C
C    CLOSE NEW PLOTTING FILES
C
      CALL COMO$$(:000001,0,0,0,I1)
      CALL COMI$$('PPSUB',5,6,I1)
C
C   RUN FILE 'PPSUB' AS A COMPUTER INPUT FILE
C
      CALL EXIT
102   CALL TNOUA(' **** File ',11)
      CALL TNOUA(IFIL,6)
      LOGY=YSNO$A(' does not exist, is entry correct ',34,A$DNO)
      IF(LOGY) CALL TNOU('Entry accepted, dummy file created',34)
      IF(.NOT.LOGY) CALL TNOU('Entry rejected',14)
      CALL TNOU('Continue.........',17)
      IF(LOGY) CALL SRCH$$(K$WRIT,IFIL,6,15,I1,I2)
      IF(LOGY) CALL SRCH$$(K$CLOS,0,0,15,I1,I2)
      IF(LOGY) GOTO 10
      GOTO 2
      END
```

Appendix 11.

Modification of PAFEC routine D61300 to write out stiffness matrix.

```
      SUBROUTINE D61300
      COMMON/IBASE/IBASE(1000)
      COMMON BASE(100)
      IF(IBASE(54).EQ.0) CALL TITLE(7)
      IM=IBASE(13)
      IF(IM.NE.2.AND.IM.NE.3) CALL D15511(IM,61300)
      ICES=IBASE(14)
      CALL R15515
      CALL R09800(2,2)
      CALL R09800(3,2)
      CALL R09800(4,2)
      CALL R09800(69,1)
      IDT=IBASE(5)
      ID=IBASE(6)
      ILO=IBASE(8)
      IX=IBASE(4)
      IEF=IBASE(43)
      IDD1=(ID*(ID+3))/2
      IDD1=IDD1*2
      IT=ID+ILO+2
      IT=IT*2
      CALL R09808(5,IDD1,1,L5,JROW5,IPOS5)
      IF(IBASE(55).EQ.0) CALL R09808(6,IDT,ILO,L6,JROW6,IPOS6)
      CALL R09808(7,IT,1,L7,JROW7,IPOS7)
      CALL R09808(8,ID,ILO,LM8,JROW8,IPOS8)
      CALL R09808(77,6,ILO,L,J,I)
      IF(IBASE(55).GT.0) CALL R09800(6,1)
      IF(IBASE(55).GT.0) CALL R09807(6)
      CALL D15501(BASE(IPOS5))
      CALL R14750(ISTAR)
      IF(ISTAR.GT.IEF) GOTO 105
         DO 100 L1=ISTAR,IEF
         IBASE(12)=L1
         CALL D14001
      CALL R14700(ISTOP)
      IF(ISTOP.EQ.1) RETURN
  100 CONTINUE
  105 CONTINUE
      IF(IBASE(54).GT.0) REWIND ICES
      IBASE(39)=IBASE(43)
      CALL R09800(3,5)
      CALL R09800(4,5)
      CALL R09800(70,5)
      CALL R09800(71,5)
      CALL R09800(73,5)
      CALL R09800(77,5)
      IATR=IBASEP(7)
      IATR=IBASE(IATR+3)
      IS=IBASEP(5)
      IS=IBASE(IS+3)
      CALL R09806(5,LM5,JROW5,IS)
      CALL LJPSMO(BASE(IS),ID)
      CALL D61301(BASE(IS),BASE(IATR))
      CALL D15509(BASE(IS),ICK,IC)
      CALL R09800(5,5)
      CALL R15510(ICK,IC)
      CALL R15600
      IF(IBASE(54).EQ.0) CALL R18006
      CALL R09800(2,5)
      CALL R09800(6,4)
      CALL R09800(69,5)
      CALL R09800(7,5)
      CALL R09800(8,5)
      CALL R09800(70,5)
      CALL R09800(71,5)
      IF(IBASE(54).GT.0) CALL R08603
      RETURN
      END
      SUBROUTINE LJPSMO(S,ID)
      COMMON /IBASE/IBASE(1000)
      COMMON BASE(100)
      REAL*8 S(1)
      INTEGER*4 IPOSS(5050),I1(100),IS,I2(100)
4     FORMAT(1H ,'STIFFNESS MATRIX, LOWER TRIANGLE,D=',I4)
1     FORMAT(1H ,(D21.14))
C
C
C FIND ACTIVE FREEDOMS IN LIST CONTAINING MANY ZERO'S
C
C
      IPOS1=(ID*(ID+1)/2)+1
      IPOS2=IPOS1+ID-1
2     FORMAT(1H ,'ACTIVE FREEDOMS ',/,(E11.4))
      J=0
      DO 10 LJ1=IPOS1,IPOS2
      IF(S(LJ1).EQ.0.0)GOTO 10
      J=J+1
      I1(J)=LJ1-IPOS1+1
      I2(J)=LJ1
10    CONTINUE
      WRITE(6,6)J,ID,IPOS1,IPOS2
C
C
C  FIND POSSITIONS OF NON-ZERO TERMS IN S.M.
C
C
      DO 20 LJ1=1,J
      DO 20 LJ2=1,LJ1
      LJ3=(LJ1*(LJ1-1))/2+LJ2
20    IPOSS(LJ3)=(I1(LJ1)*(I1(LJ1)-1))/2+I1(LJ2)
C
C
C   GET DOF NODE DIRECTION INFO AND FIND POSITION IN BASE
C
C
      CALL R09800(2,1)
      CALL R09806(2,LM2,JROW2,IPOSS2)
      IBEG2=IPOSS2
      IEND2=IPOSS2+LM2-1
C
C
C   WRITE OUT  ACTIVE FREEDOMS AS INTEGERS
C
C
      WRITE(5,6)J
6     FORMAT(I5)
      DO 7 L9=1,J
      IS=S(I2(L9))
7     WRITE(5,6) IS
C
C
C    WRITE OUT SIZE OF S.M. THEN WRITE OUT S.M.
C
C
      WRITE(5,1)(S(IPOSS(L9)),L9=1,LJ3)
C
C
C    FIND NODES AND DIRECTIONS RELEVENT TO ACTIVE FREEDOMS
C
C
      DO 5 II=IBEG2,IEND2
      IDOF=BASE(II)
      IF(IDOF.EQ.0) GOTO 5
      INODE=FLOAT(II)/JROW2+0.9
      IDIRE=II-(INODE-1)*JROW2
      WRITE(5,3) IDOF,INODE,IDIRE
3     FORMAT(3I10)
5     CONTINUE
      CALL TNOU(' JOB STOPPED AT THE END OF REDUCTION PHASE ',42)
      STOP 200
      END
```